

CR-172001

DOC. TYPE 2  
MGT. CODE OM

SISO-UG-001

## ★ USER'S GUIDE

# CALIBRATED ANCILLARY SYSTEM (CAS)

## VOLUME IX

- CAS Commands
- Acronyms and Abbreviations
- Applicable Documentation

PREPARED BY



Ford Aerospace &  
Communications Corporation  
Space Information Systems Operation

CONTRACT NAS 9-15014  
DRL LI 81

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER

(NASA-CR-172001) CALIBRATED ANCILLARY  
SYSTEM (CAS) USER'S GUIDE, VOLUME 9 (Ford  
Aerospace and Communications Corp.) 209 P  
Avail: NTIS HC A10/HF A01 CSCI 09B

N87-28293

Unclass  
G3/b1 0092947

31 MAY 1986

# CONTENTS

This user's guide contains the introduction to the equipment, operation, general procedures, and specific procedures of the calibrated ancillary system (CAS).

## Volume I

### Introduction

- Chapter 1 CAS Overview
- Chapter 2 User Categories
- Chapter 3 Introduction to Operation
- Chapter 4 General Operating Procedures

## Volume II

- Chapter 5 Data Flow Engineer Procedures

## Volume III

- Chapter 5 Data Flow Engineer Procedures (Cont.)

## Volume IV

- Chapter 6 GSFC User Mission Planning Procedures.

## Volume V

- Chapter 6 Testing User Mission Planning Procedures

## Volume VI

- Chapter 6 Testing User Mission Planning Procedures (Cont.)

## Volume VII

- Chapter 6 Data Flow Engineer Mission Planning Procedures

## Volume VIII

- Chapter 7 Checkout and Software Procedures
- Chapter 8 Maintenance User Procedures
- Chapter 9 System Manager Procedures

## Volume IX

- Appendix A CAS Commands
- Appendix B Acronyms and Abbreviations
- Appendix C Applicable Documentation

APPENDIX A  
CAS COMMANDS

APPENDIX B  
ACRONYMS AND  
ABBREVIATIONS

APPENDIX C  
APPLICABLE  
DOCUMENTATION

## APPENDIX A

### CALIBRATED ANCILLARY SYSTEM COMMANDS

The commands listed here can be executed no matter what menu is visible to the user. Each command will be followed by a parenthesis giving the short form of the command. When a command is entered without its associated parameters the user will be prompted for information to complete the command. An example of the command and its parameters will also be presented.

### ANCILLARY

#### 1. ancstart (ancst)

This initializes the ancillary process. You will be prompted for the SDSS port, the flight ID for stream 1 & 2, the data stream tag, the revision code for the support files and the NDD # where the data is coming from.

#### Example:

```
ancst 0 51-e rt a 9 DONE d d d
      | | | | | | | | | |
      1 2 3 4 5 6 7 8 9
      stream 1 stream 2
```

This example shows the initialization of a single stream for the ancillary process. Parameters 6-9 are necessary for the command to be accepted.

```
1 SDSS port
2 & 6 Flight ID
3 & 7 Data Stream tag (rt,
  sim, pb)
4 & 8 Revision code (a - z)
5 & 9 NDD # (1 - 4 or 9)
```

#### 2. ancrestart (ancr)

This command restarts the ancillary process if it was started with the wrong set of support files, if you received error messages, if you want to use a different set of support files or change the NDD you want to listen to. This command will restart ancillary without stopping the ancillary process and then initialize it again.

## Example:

ancr 0 51-e rt a 9 DONE d d d

3. anchalt (anch)

This command will halt the ancillary process. No other parameters are needed.

## Example:

anch

4. ancoutenable (ancoutel)

This enables the output of the ancillary process to the SDSS. You will be prompted for the stream you want to enable.

## Example:

ancoute 1

| This parameter is stream number, valid values are 1 or 2.

5. ancoutdisable (ancoutd)

This disables the output of the ancillary process to the SDSS. You will be prompted for the stream you want disabled.

ancoutd 1  
 Example:

ancoutd 1

| This parameter is stream number, valid values are 1 or 2.

6. ancsend (ancse)

The ancillary send health and status message, command the ancillary process to send a message to health and status. There are no other parameters needed.

Example:

ancse

NDD1. nddappstart (nddappsta)

This initializes the NDD application. You will be prompted for flight number, data base version, data stream tag, and operational mode.

Example:

```
nddappsta 51-e ^ a rt ops
          | | | | |
          1 2 3 4
```

- 1 Flight ID
- 2 Data base version (a - z)
- 3 data stream tag (rt, sim  
or pb)
- 4 Operational mode (ops or dev)

2. nddhalt (nddh)

This command halts the NDD application. No other parameters are needed.

Example:

```
nddh
```

3. selwindow (selw)

The select window command will only operate if the flag DEBUG is set in the module output dcp.c and recompiled. The when the NDD application is initialized the dev mode is used. When this is done a display showing the status bits of the minor frames is exhibited. The user will be prompted for the window number.

Example:

```
selw 1
      | Valid window numbers are 1-7.
```

LOGGING

\*\*\* Pre -initialization \*\*\*

Pre-initialization is a action whereby the user will determine the characteristics of the log files he will be logging to. The user will determine the number of log files, the data type, the log file size, the log file name, logging file type and if the logging process will switch automatically from one log file to another. Once the command is given the user will be prompted for the above information. After giving the necessary information, enter the log enable command for logging to commence.

## LOGGING PARAMETERS AND THEIR VALUES

In this section the parameters and their valid values are listed below:

APPLICATION	PROCESS
ndd	ndd
ancillary	ancillary
support	health_stat user_if genlog supervisor lan_if
checkout	data_generator score scriptor



DATA TYPE	value	maximum log file size in megabytes	default log name
User Interface commands	1	.5	USER_IF
Health and Status messages	2	1	HS_MSG
CSAC messages	3	1	CSAC-MSG
DCP	4	5	DCP
NASCOM Blocks	5	5	NASCOM
OD Minor Frames	6	5	OD FRAME
Data generator DCP's	7	5	T_DCP
Data generator NASCOM	8	5	T_NASCOM
Data generator OD Frames	9	5	T_OD_FRM
Utility (HWC0 Data)	10	.5	
SDSS loop back NASCOM	11	5	L_NASCOM

FILE SIZE - The number of bytes given as an integer value. The chart above gives the maximum size allowed for each file initialized. If the user uses the value "0" logger will default to the maximum log file size.

FILE NAME - log file name is limited to 14 characters.

DELOG TITLE - the delog title is limited to 40 characters.

LOG FILE DEVICE - is either floppy (0) or disk (1).

SWITCH AT END OF FILE - if the user chooses not to switch at the end of file, circular logging will occur. The values for this parameter are "0" for no and "1" for yes.

#### 1. loginit (logi)

This command will pre-initialize log files which the user will name. The following information will be prompted for: application, process, logging data type, log file device, number of files, log file name, maximum log file size, switch at end of file. The information log file name through switch at end of file will be prompted for twice even if you only indicate that you only want one log file pre-initialized. A command string with all the parameters can not be input on the control line since the input line is too large; an example of how this command can be used in a batch file is given. This command must be followed by a loge (log enable) command for logging to commence.

## Example:

```
logi
nnd
nnd
4
1
1
nnd.log1
20000
nnd.dlog1
1
\0
0
\0
1
(application)
(process)
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```

In the example above the number of log files to be pre-initialized was 1. Even though only one file is to be pre-initialized, defaults for the second log file must be given in the command.

## 2. ancloginit (anclogi)

This command will set up preinitialized log files for the ancillary process. The user will be prompted for logging data type, log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the anclog command to start logging. The command line for this command is too long to be input on the control line: however, an example of how this command can be used in a batch file is given.

### Example:

```
anclogi
4
1
1
nld.log1
200000
nld.dlog1
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```

### 3. csacloginit (csaclogi)

This command will set up preinitialized log files for the csac process. The user will be prompted for log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the csaclog command to start logging. The command line for this command is too long to be input on the control line; however, an example of how this command can be used in a batch file is given.

```

Example:
csaclogi
4          (data type)
1          (log file device)
1          (# of log files)
ndd.log1  (log file name, user's decision)
200000    (log file size)
ndd.dlog1 (delog title )
1          (switch at end of file)
\0        (2nd log file name)
0         (2nd log file size)

\0        (2nd delog title)
1         (2nd log switch at end of log file)

```

#### 4. dgloginit (dglogi)

This command will set up preinitialized log files for the data generator process. The user will be prompted for logging data type, log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the dgloge command to start logging. The command line for this command is too long to be input on the control line; however, an example of how this command can be used in a batch file is given.

#### Example:

```
dglogi
4
1
1
nnd.log1
20000
nnd.dlog1
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```

5. gloginit (glogi)

This command will preinitialize general log files which the user will name. The user will be prompted for the logfile device, number of log files, the names of the log files, the size of the files, the delog titles for each file and if a switch should occur at the end of the file. If the user indicates only one file he will still be prompted for information regarding the second file. The command line is too long to enter on the control line of the screen therefore no example is given; however, an example of how this command can be used in a batch file is given.

## Example:

```

glogi
4
1
1
ndd.log1
20000
ndd.dlog1
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)

```

## 6. hsloginit (hslogi)

This command will preinitialize health and status log files which the user will name. The user will be prompted for the logfile device, number of log files, the names of the log files, the size of the files, the delog titles for each file and if a switch should occur at the end of the file. If the user indicates only one file he will still be prompted for information regarding the second file. The command line is too long to enter on the control line of the screen therefore no example is given; however, an example of how this command can be used in a batch file is given.

### Example:

```
hslogi
4
1
1
nndd.log1
200000
nndd.dlog1
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```



## 7. nddloginit (nddlogi)

This command will set up preinitialized log files for the NDD process. The user will be prompted for logging data type, log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the nddloge command to start logging. The command line for this command is too long to be input on the control line; however, an example of how this command can be used in a batch file is given.

Example:

```
nddlogi
4
1
1
ndd.log1
20000
ndd.dlog1
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```

8. ndgloginit (ndglogi)

This command will set up preinitialized log files for the data generator process. The user will be prompted for logging data type, log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the ndglog command to start logging. The command line for this command is too long to be input on the control line; however, an example of how this command can be used in a batch file is given.

## Example:

```
ndglogi
4          (data type)
1          (log file device)
1          (# of log files)
nnd.log1  (log file name, user's decision)
200000    (log file size)
nnd.dlog1 (delog title )

1          (switch at end of file)
\0         (2nd log file name)
0          (2nd log file size)
\0         (2nd delog title)
1          (2nd log switch at end of log file)
```

## 9. scologinit (scologqi)

This command will set up preinitialized log files for the scoring process. The user will be prompted for logging data type, log file device, number of log files, the delog titles, the log file sizes, and if the user wants to switch log files. This command must be followed by the scologe command to start logging. The command line for this command is too long to be input on the control line; however, an example of how this command can be used in a batch file is given.

### Example:

```
scologi
4
1
1
ndd.logl
20000
ndd.dlogl
1
\0
0
\0
1
(data type)
(log file device)
(# of log files)
(log file name, user's decision)
(log file size)
(delog title )
(switch at end of file)
(2nd log file name)
(2nd log file size)
(2nd delog title)
(2nd log switch at end of log file)
```

10. uifloginit (uiflogi)

This command will pre-initialize log files for user interface. The user will be prompted for log file device, number of log files, log file names, log file sizes, and if to switch at end of file. The input line is too long to enter therefore there will be no example; however, an example of how this command can be used in a batch file is given.

## Example:

```
uiflogi
4      (data type)
1      (log file device)
```

```
1      (# of log files)
nnd.log1 (log file name, user's decision)
200000   (log file size)
nnd.dlog1 (delog title )
1        (switch at end of file)
\0       (2nd log file name)
0        (2nd log file size)
\0       (2nd delog title)
1        (2nd log switch at end of log file)
```

### \*\*\* Log enable \*\*\*

The log enable command will commence logging to a log file. If no log file has been pre-initialized logging will be directed to a default log file for a particular data type dependent on the process which logging is occurring for. The user will be prompted for the necessary information.

#### 11. logenable (loge)

This command will enable the logging for a specific data type. The following information will be prompted for: application, process, logging data type, and logging file type. If this command is used without the loginit command logging will be directed to a default log file whose name has been predetermined by the data type. The following is a list of the default file names and the corresponding data types:

Data type	Default Log File Name
1. DCP	DCP
2. Data Generator DCP's	T_DCP
3. OD Frames	OD_FRAME
4. Data Generator OD	T_OD_FRM
5. NASCOM Blocks	NASCOM
6. Data generator	T_NASCOM
NASCOM Blocks	
7. Health & Status	HS MSG
8. User Interface	USER_IF
9. CSAC Messages	CSAC_MSG
10. General Default	GENERAL
Log File	

```

Example:
logc ndd ndd ^ 4 1
      |      | ^  |  |
      1      2 ^  3  4
1 Application ( support, ndd, ancillary, or
               checkout)
2 Process (health_stat, user_if, ndd,
  ancillary, data_generator, score, or
  scriptor)

3 Data type ( 1 User Interface Commands
              ( 2 Health and Status Messages
              ( 3 CSAC Messages
              ( 4 Data Cycle Packets
              ( 5 NASCOM Blocks
              ( 6 OD Minor Frames
              ( 7 DCP's from Data Generator
              ( 8 NASCOM Blocks from Data
                Generator
              ( 9 OD Minor Frames from Data
                Generator
              (10 Utility
              (11 SDSS Loop back NASCOM

4 Logging file type ( 0 General Log file
                    ( 1 Data Type Specific
                      ( Log File

```

12. anclogenable (ancloge)

This command enables the logging of the output of the ancillary process. If this command is given without the anclogi command logging will be directed to a default file. The user will be prompted for logging data type and the logging file type.

## Example:

```
ancloge 4 1 ^
          | |
          1 2

          1 Data type (4 - DCP
            (5 - NASCOM Blocks)

          2 Logging file type ( 0 - General
                                ( 1 - Data Specific)
```

13. csaclogenable (csacloge)

This command will enable the logging of csac messages. The user will be prompted for the logging file type.

## Example:

```
csacloge 1      (0 - general log file)
              (1 - data type specific)
```

14. dglogenable (dgloge)

This command will enable the logging of data generator data. This command should be used on the workstation only. The user will be prompted for the data type and the logging file type.

## Example:

```
dgloge 7 1 ^
          | |
          1 2

          1 Data type (7 - data cycle packets)
            (8 - NASCOM Blocks)

          2 Logging file type (0 - general log file)
                                (1 - data type specific)
```

15. hslogenable (hsloge)

This command will enable the logging of health and status messages. The user will be prompted for the logging file type.

## Example:

```
hsloge 1      (0 - general log file)
              (1 - data type specific)
```

16. nddlogenable (nddloge)

This command will enable the logging of the specified data type within the NDD process. The user will be prompted for the data type and the logging file type.

## Example:

```
nddloge 4 1 1 data type (4 - Data Cycle Packets)
          | | (6 - OD Frames)
          1 2 2 logging file type (0 - general log file)
              (1 - data type specific)
```

17. ndglogenable (ndgloge)

This command will enable the logging of data generator data. This command should be given on an NDD only. The user will be prompted for the data type and the logging file type.

## Example:

```
ndgloge 7 1 1 Data type (7 - Data Cycle Packets)
          | | (9 - OD Frames)
          1 2 2 Logging file type (0 - general log file)
              (1 - data type specific)
```



18. scologenable (scologe)

This command will enable the logging of scoring process. The user will be prompted for the logging file type.

Example:

```
scologe 1      (0 - general log file)
              (1 - data type specific)
```

19. uiflogenable (uifloge)

This command will enable the logging of user interface commands. The user will be prompted for the logging file type.

Example:

```
uifloge 1      (0 - general log file)
              (1 - data type specific)
```

\*\*\* Disable logging \*\*\*

The disable logging command will pause logging of a particular data type from a particular process. The user will be prompted for the necessary information.

20. logdisable (logd)

This command will disable logging of a specific data type. The information prompted will be the following: application, process, and logging data type.

## Example:

```
logd ^ ndd ndd 4 ^
| | | |
1 2 3
3 Data type ( 1 User Interface Commands )
( 2 Health and Status Messages )
( 3 CSAC Messages )
( 4 Data Cycle Packets )
( 5 NASCOM Blocks )
( 6 OD Minor Frames )
( 7 DCP's from Data Generator )
( 8 NASCOM Blocks from Data )
( Generator )
( 9 OD Minor Frames from Data )
( Generator )
(10 Utility )
```

21. anclogdisable (anclogd)

This command disables the logging of a specified data type of the ancillary process. The user will be prompted for the logging data type.

## Example:

```
anclogd 4 ( 4 - DCP )
( 5 - NASCOM Block )
```

## 22. csaclogdisable (csaclogd)

This command disables the logging of csac messages. There are no parameters for this command.

Example:

csaclogd

## 23. dglogdisable (dglogd)

This command disables the logging of the data generator process. This command should be given only on a workstation. The user will be prompted for the logging data type.

Example:

dglogd 7 (7 - Data Cycle Packets)  
(8 - NASCOM Blocks)

## 24. hslogdisable (hslogd)

This command disables the logging of health and status messages. There are no parameters for this command.

Example:

hslogd

25. nddlogdisable (nddlogd)

This command disables the logging of a specified data type of the NDD process. The user will be prompted for the logging data type.

Example:

nddlogd 4 (4 - Data Cycle Packets)  
(6 - OD Frames)

26. ndglogdisable (ndglogd)

This command disables the logging of the data generator process. This command should be given only on an NDD. The user will be prompted for the logging data type.

Example:

ndglogd 7 (7 - Data Cycle Packets)  
(9 - OD Frames)

uiflogd

\*\*\* Re-initialization \*\*\*

The re-initialization command will be used when the user obtains an error message indicating that the logging function for a particular data type is unavailable or when the delog table could not be accessed.

28. logreinit (logrei)

```
Example:
```

logrei	n dd	n dd				
	1	1	2	2	2	2

	Application Process
1 Application	1 Process
2 Application	2 Process

29. anclogreinit (anclogrei)

**Example:**

**anclogrei**

30. csaclogreinit (csaclogrei)

This command re-initializes the logging function of csac messages. There are no parameters for this command.

Example:

csaclogrei

31. dglogreinit (dglogrei)

This command re-initializes the logging function of the data generator. There are no parameters for this command.

Example:

dglogrei

32. glogreinit (glogrei)

This command re-initializes the general logger. There are no parameters for this command.

Example:

glogrei

33. hslogreinit (hslogrei)

This command re-initializes the logging function of health and status messages. There are no parameters for this command.

Example:

hslogrei

34. nddlogreinit (nddlogrei)

This command re-initializes the logging function of the NDD process. There are no parameters for this command.

Example:

nddlogrei

35. scollogreinit (scollogrei)

This command re-initializes the logging function of the scoring process. There are no parameters for this command.

Example:

scollogrei

36. uiflogreinit (uiflogrei)

This command re-initializes the logging function of the user interface process. There are no parameters for this command.

Example:

uiflogrei

**\*\*\* Log switch \*\*\***

The log switch command will redirect logging of a data type from one data specific log file to another, or from a data specific log file to the general log file, or from a general log file to a data specific log file. The user will be prompted for either data type and/or logging file type. Once this command is given the log file which had been logged to will be closed and moved from the active\_files directory to the log\_files directory where it can be delogged.

**37. logswitch (logs)**

This command will switch logging from data specific to general log file or from general log file to data type specific log file. You will be prompted for the following information: application, process, logging data type, type of log file logging will be switched from, and the type of log file switched to.



## Example:

```

logs ndd ndd 4 1 1
  |  |  |  |  |
  1  2  3  4  5

1 Application ( support, ndd, ancillary, or
  checkout)
2 Process (health_stat, user_if, ndd, or
  ancillary)
3 Data type ( 1 User Interface Commands
  ( 2 Health and Status Messages
  ( 3 CSAC Messages
  ( 4 Data Cycle Packets
  ( 5 NASCOM Blocks
  ( 6 OD Minor Frames
  ( 7 DCP's from Data Generator
  ( 8 NASCOM Blocks from Data
  ( Generator
  ( 9 OD Minor Frames from Data
  ( Generator
  (10 Utility
4 Log file switching from (0 General Log file)
  ( 1 Data Specific Log
  File
5 Log file switching to (0 General Log file)
  (1 Data Specific Log
  File

```

38. anclogswitch (anclogs)

This command will switch data type specific log files of the specified data type within the ancillary process. The user will be prompted for the logging data type and when the switch should occur.

## Example:

```
anclogs 4 1
         ^ |
         | |
         1 2
```

```
1 Data type (4 - DCP
             (5 - NASCOM Blocks)
```

```
2 When Switch should occur (0 - Immediately)
                             (1 - At end of
                               current file)
```

39. anc2logswitch (anc2logs)

This command will switch logging of a data type within the ancillary process from data specific log files to general log files. The user will be prompted for the logging data type. The switch will occur immediately.

## Example:

```
anc2logs 4      (4 - DCP
                 (5 - NASCOM Blocks)
```

This command will switch logging of a data type within the ancillary process from general log files to data specific log files. The user will be prompted for the logging data type. The switch will occur immediately.

40. anc3logswitch (anc3logs)

Example:

```
anc3logs 4      (4 - DCP
                 (5 - NASCOM Blocks)
```

41. csac1logswitch (csac1logs) This command will switch csac data specific log files. The user will be prompted for when to switch the log files.

Example:

```
csac1logs 0      (0 - immediately )
                (1 - at end of file )
```

42. csac2logswitch (csac2logs) This command will switch csac data specific log file to general log file. There are no parameters for this command.

Example:

```
csac2logs
```

43. csac3logswitch (csac3logs)

This command will switch logging of CSAC messages from the general log file to data specific log file. There are no other parameters for this command.

Example:

csac3logs

44. dglllogswitch (dglllogs)

This command will switch logging of a data type within the data generator process. The user will be prompted for the logging data type and when the switch should occur. This command should only be given in a workstation.

Example:

```
dglllogs 7 1 ^
          | |
          1 2

          1 Data type (7 - DCP
            (8 - NASCOM Blocks)

          2 When Switch should occur (0 - Immediately)
            (1 - At end of
              file
            )
```

45. dg2logswitch (dg2logs)

This command will switch logging of a data type within the data generator process from data specific log files to general log files. The user will be prompted for the logging data type. The switch will occur immediately. This command should only be given in a workstation.

Example:

```
dg2logs 7      (7 - DCP  
                (8 - NASCOM Blocks)
```

46. dg3logswitch (dg3logs)

This command will switch logging of a data type within the data generator process from general log files to data specific log files. The user will be prompted for the logging data type. The switch will occur immediately. This command should only be given in a workstation.

Example:

```
dg3logs 7      (7 - DCP  
                (8 - NASCOM Blocks)
```

#### 47. glogswitch (glogs)

This command will switch general log files. The user will be prompted for when the switch should occur.

Example:

```
glogs 0      0 Immediately
          1 End of current file
```

#### 48. hsllogswitch (hsllogs)

This command will switch health and status data specific log files. The user will be prompted for when to switch the log files.

Example:

```
hsllogs 0    (0 - immediately )
            (1 - at end of file )
```

#### 49. hs2logswitch (hs2logs)

This command will switch health and status data specific log file to general log file. There are no other parameters for this command.

Example:

hs2logs

#### 50. hs3logswitch (hs3logs)

This command will switch health and status general log file to data specific log file. There are no other commands for this command.

Example:

hs3logs

51. nddllogswitch (ndqllogs)

This command will switch logging of a data type within the NDD process. The user will be prompted for the logging data type and when the switch should occur.

Example:

```
nddllogs 4 1 ^
          | |
          1 2
          1 Data type (4 - DCP          )
            (6 - OD Frames          )
          2 When Switch should occur (0 - Immediately)
            (1 - At end of          file          )
```

52. ndd2logswitch (ndd2logs)

This command will switch logging of a data type within the NDD process from data specific log files to general log files. The user will be prompted for the logging data type. The switch will occur immediately.

Example:

```
ndd2logs 4 (4 - DCP          )
           (6 - OD Frames          )
```

53. ndd3logswitch (ndd3logs)

This command will switch logging of a data type within the NDD process from general log files to data specific log files. The user will be prompted for the logging data type. The switch will occur immediately.

Example:

```
ndd3logs 4 (4 - DCP )
          (6 - OD Frames )
```

54. ndgllogswitch (ndgllogs)

This command will switch logging of a data type within the data generator process. The user will be prompted for the logging data type and when the switch should occur. This command should only be given only in an NDD.

Example:

```
ndgllogs 7 1 ^
          | |
          1 2

          1 Data type (7 - DCP )
            (9 - OD Frames )

          2 When Switch should occur (0 - Immediately)
            (1 - At end of current file)
```



55. ndg2logswitch (ndg2logs)

This command will switch logging of a data type within the data generator process from data specific log files to general log files. The user will be prompted for the logging data type. The switch will occur immediately. This command should only be given in an NDD.

Example:

```
ndg2logs 7 (7 - DCP
           (9 - OD Frames )
```

56. ndg3logswitch (ndg3logs)

This command will switch logging of a data type within the data generator process from general log files to data specific log files. The user will be prompted for the logging data type. The switch will occur immediately. This command should only be given in an NDD.

Example:

```
ndg3logs 7 (7 - DCP
           (9 - OD Frames )
```

57. scollogswitch (scollogs)

This command will switch log files for the scoring process. The user will be prompted for when the switch should occur.

Example:

```
scollogs 0
```

When Switch should occur (0 - Immediately)  
(1 - At end of  
file )

58. sco2logswitch (sco2logs)

This command will switch log files of the scoring process from data specific log files to general log files. The switch will occur immediately. There are no other parameters for this command.

Example:

sco2logswitch

59. sco3logswitch (sco3logs)

This command will switch log files of the scoring process from general log files to data specific log files. The switch will occur immediately. There are no other parameters for this command.

Example:

sco3logs

60. uifllogswitch (uifllogs)

This command will switch user interface data specific log files. The user will be prompted for when to switch the log files.

Example:

```
uifllogs 0      (0 - immediately )
            (1 - at end of file )
```

61. uif2logswitch (uif2logs) This command will switch user interface data specific log file to general log file. There are no other parameters for this command.

Example:

uif2logs

62. uif3logswitch (uif3logs) This command will switch user interface general log file to data specific log file. There are no other parameters for this command.

Example:

uif3logs

\*\*\* Halt \*\*\*

The halt command will disable the logging of all data types within a process which logs multiple data types.

63. ancloghalt (anclogh) This command halts the logging of the output of the ancillary process. There are no parameters associated with this command.

Example:

anclogh

#### 64. dgloghalt (dglogh)

This command halts the logging of the data generator process. There are no parameters associated with this command.

Example:

dglogh

#### 65. ghaltproc (ghaltp)

This command will halt the general logger. No information will be prompted for.

Example:

ghaltp

#### 66. nddloghalt (nddlogh)

This command halts the logging of the NDD process. There are no parameters associated with this command.

Example:

nddlogh

DELOG

\*\*\*Debugger Process\*\*\*

1. dloghaltproc (dloghaltp)

The delog halt process will halt the delog process not just a delog session. No other parameters are required.

Example:

dloghaltp

2. dlogstart (dlogsta)

The delog start command will start the delog process, this command should only be issued once before requesting delog sessions. This command will only start the delog process not a delog session. No other parameters are required.

Example:

dlogsta

3. dlogreinit (dlogrei)

The delog reinitialization command should be used when debugger returns an error condition. If this still does not correct the problem then development should be called. there are no other parameters used with this command.

Example:

dlogrei

**\*\*\*Delog Session\*\*\***

4. dloghaltcurrent (dloghaltc) The delog halt current session will halt the present delog session only not the delog process. No other parameters are required.

Example:

dloghaltc

5. dloghaltall (dloghalta)

The delog halt all current sessions will cancel all delog sessions that you have queued up. It will not halt the delogger process its self. There are no other parameters needed.

Example:

dloghalta

The request `delog` session command will allow the user to enter a `delog` session. He will be prompted for log file device, log file name, the `delog` listing device, the format for the `delog`, the `delog` file name if device is disk, the `delog` format, the data type of the data, `delog` by time the `delog` clock, the start and stop time.

```
dlogreq 2 logl 2 0 delog1 2 4 0 0 beg end      1 - log file device  
          |   |   |   |   |   |   |           (0 - floppy )  
          |   |   |   |   |   |   |           (2 - disk  )  
  
          1   2   3   4       5               6   7   8   9    10  11  
2 - log file name      3 - delog listing device  
  (this requires )      (0 - line printer)  
  (the user to )        (1 - CRT )  
  (enter the name)      (2 - disk  )  
  (of the log file)  
  (being delogged )  
  (up to 15 chars )  
  (if the value of parameter 1 is 0)  
  (then the value here is \0 )
```

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - data type
  (0 -all types )
  (1 - user interface )
  (2 - health & status )
  (3 - CSAC )
  (4 - DCP's )
  (5 - NASCOM Blocks )
  (6 - OD Frames )
  (7 - DCP's from Data Generator )
  (8 - NASCOM Blocks from Data Generator )
  (9 - OD Frames from Data Generator )
  (10 - Utility )

8 - delog by time
  (0 - No )
  (1 - Yes )

9 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

10 - start time
  (format is )
  (ddd:hh:mm:ss.sss or )
  (the value beg if not)
  (delogging by time )

11 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

```



**Example:**

A-47

- 6 - delog format
  - (0 - hex dump )
  - (1 - ascii dump )
  - (2 - predefined format )
  - (3 - headers only )
  - (4 - summary only )
- 7 - delog by time
  - (0 - No )
  - (1 - Yes )
- 8 - delog clock
  - (0 - GMT clock )
  - (1 - Alternate clock )
- 9 - start time
  - (format is ddd:hh:mm:ss.sss or )
  - (the value is beg if not delogging)
  - (by time.)
- 10 - stop time
  - (format is ddd:hh:mm:ss.sss or )
  - (the value end if not delogging )
  - (by time )

8. csacdlogrequest (csacdlogreq) The csac delog request command will allow the user to enter a csac delog session, and delog only csac data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process. This command will delog all message types which are classified as csac.

The message types are:

1. Initialization messages
2. Advisory messages
3. Error messages
4. General status messages
5. Interface messages
6. Operating System status messages
7. Queue status messages
8. Configuration messages
9. Acknowledgement messages
10. Restart messages
11. Verification messages
12. Test messages
13. Process termination messages
14. Report available
15. CSAC status

**Example:**

```

csacdlogreq 2 logl 2 0 0 delog1 2 0 0 beg end -1 -1      1 - log file device
              |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
              1   2   3   4   5       6   7   8   9    10  11 12
2 - log file name
  (this requires )
  (the user to )
  (enter the name)
  (of the log file)
  (being delogged )
  (up to 15 chars )
  (if the value of parameter 1 is 0)
  (then the value here is \0 )

```

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

A-51

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

10. csac2dlogrequest (csac2dlogreq) The csac delog two request command will allow the user to enter a csac delog session, and delog only csac data type which are advisory messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if

device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

**Example:**

[illegible]

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

```

```

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



**Example:**

A-55

```

8 - delog clock          9 - start time
  (0 - GMT clock        ) (format is ddd:hh:mm:ss.sss or )
  (1 - Alternate clock ) (the value is beg if not delogging)
                           ) (by time.)

10 - stop time          11 - originating application
  (format is ddd:hh:mm:ss.sss or ) (support )
  (the value end if not delogging ) (checkout )
  (by time                  ) (ndd )
                           ) (ancillary )
                           ) (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

12. csac4dlogrequest (csac4dlogreq) The csac delog four request command will allow the user to enter a csac delog session, and delog only csac data type which are general status messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

**Example:**

```

csac4dlogreq 2 logl 2 0 delog1 2 0 0 beg end -1 -1 1 - log file device
              |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
              1   2   3   4   5   6   7   8   9   10  11  12
2 - log file name
  (this requires )
  (the user to )
  (enter the name)
  (of the log file)
  (being delogged )
  (up to 15 chars )
  (if the value of parameter 1 is 0)
  (then the value here is \0 )

```

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-l all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

## CAS User's Guide

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

- name if device is disk, the delog format,  
delog by time the delog clock, the  
start,stop time, originating application  
and originating process.

[illegible]

(if the value of parameter 1 is 0)  
(then the value here is \0)

A-61

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



**Example:**

**A-63**

```

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

16. csac8dlogrequest (csac8dlogreq) The csac delog eight request command will allow the user to enter a csac delog session, and delog only csac data type which are configuration messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

**Example:**

```

csac8dlogreq 2 logl 2 0 delogl 2 0 0 beg end -1 -1 ^
              |   |   |   |   |   |   |   |   |
              1   2   3   4   5   6   7   8   9   10  11  12
2 - log file name
  (this requires )
  (the user to )
  (enter the name)
  (of the log file)
  (being delogged)
  (up to 15 chars)
  (if the value of parameter 1 is 0)
  (then the value here is \0 )

1 - log file device
  (0 - floppy )
  (2 - disk )

3 - delog listing device
  (0 - line printer)
  (1 - CRT )
  (2 - disk )

```

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or
  (the value is beg if not delogging)
  (by time.

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

17. csac9dlogrequest . (csac9dlogreq) The csac delog nine request command will allow the user to enter a csac delog session, and delog only csac data type which are acknowledgement messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

**Example:**

[illegible]

```

2 - log file name
    (this requires )
    (the user to )
    (enter the name).
    (of the log file)
    (being delogged )
    (up to 15 chars )
    (if the value of parameter 1 is 0)
    (then the value here is \0 )

3 - delog listing device
    (0 - line printer)
    (1 - CRT )
    (2 - disk )

```

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (nnd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

- Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (nnd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



**Example:**

A-71

```

6 - delog format
(0 - hex dump )
(1 - ascii dump )
(2 - predefined format )
(3 - headers only )
(4 - summary only )

7 - delog by time
(0 - No )
(1 - Yes )

8 - delog clock
(0 - GMT clock )
(1 - Alternate clock )

9 - start time
(format is ddd:hh:mm:ss.sss or )
(the value is beg if not delogging)
(by time.)

10 - stop time
(format is ddd:hh:mm:ss.sss or )
(the value end if not delogging )
(by time

11 - originating application
(support )
(checkout )
(ndd )
(ancillary )
(-1 all applications)

12 - originating process
(health_stat )
(user_if )
(lan_if )
(ancillary )
(ndd )
(data_generator )
(descriptor )
(score )
(Delgger )

```

**Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or
   (the value is beg if not delogging)
   (by time.

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-l all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

```

2 - log file name
  (this requires )
  (the user to )
  (enter the name)
  (of the log file)
  (being delogged )
  (up to 15 chars )
  (if the value of parameter 1 is 0)
  (then the value here is \0 )

3 - delog listing device
  (0 - line printer)
  (1 - CRT )
  (2 - disk )

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the )
  (user to enter the )
  (file up to 15 char )
  (device is disk, if )
  (then the entry on )
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

22. csacl4dlogrequest (csacl4dlogreq) The csac delog fourteen request command will allow the user to enter a csac delog session, and delog only csac data type which are report available messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file

name if device is disk, the delog format,  
delog by time the delog clock, the  
start,stop time, originating application  
and originating process.

**Example:**

```
csacl4dlogreq 2 logl 2 0 delogl 2 0 0 beg end -1 -1 1 - log file device  
| | | | | | | | | | | | | | | |  
1 2 3 4 5 6 7 8 9 10 11 12  
2 - log file name  
    (this requires )  
    (the user to )  
    (enter the name)  
    (of the log file)  
    (being delogged )  
    (up to 15 chars )  
    (if the value of parameter 1 is 0)  
    (then the value here is \0 )
```

```
3 - delog listing device  
    (0 - line printer)  
    (1 - CRT )  
    (2 - disk )
```

```
4 - output format      5 - delog file name  
    (0 - 80 column )   (this requires the user to enter the)  
    (1 - 132 column)   (user to enter the name of the delog)  
                        (file up to 15 characters if the )  
                        (device is disk, if it is not disk )  
                        (then the entry on the command string)  
                        (should be \0 )
```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - NO )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health stat )
  (user if )
  (lan if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



23. csac15dlogrequest (csac15dlogreq)

logreg) The csac delog fifteen request command will allow the user to enter a csac delog session, and delog only csac data type which arecsac status messages. He will be prompted for log file device, the file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

**Example:**

```

csac15dlogreq 2 log1 2 0 delog1 2 0 0 beg end -1 -1 1 - log file device
(0 - floppy )
(2 - disk )

```

```
2 - log file name
3 - delog listing device
```

```
(this requires )
(the user to )
(enter the name)
(of the log file)
(being delogged)
(up to 15 chars)
(if the value of
(then the value h
```

```
3 - delog listing device
(0 - line printer)
(1 - CRT )
(2 - disk )
```

4 - output format  
(0 - 80 column )  
(1 - 132 column)

```
5 - delog file name
    (this requires the user to enter the)
    (user to enter the name of the delog)
    (file up to 15 characters if the )
    (device is disk, if it is not disk )
    (then the entry on the command string)
    (should be \0 )
```

6 - deloq format

7 - delog by time  
(0 - No )  
(1 - Yes )

```
(0 - hex dump
(1 - ascii dump
(2 - predefined format
(3 - headers only
(4 - summary only
)
```

```

8 - delog clock          9 - start time
  (0 - GMT clock        ) (format is ddd:hh:mm:ss.sss or )
  (1 - Alternate clock ) (the value is beg if not delogging)
                           ) (by time.)

10 - stop time          11 - originating application
  (format is ddd:hh:mm:ss.sss or ) (support )
  (the value end if not delogging ) (checkout )
  (by time                ) (ndd )
                           ) (ancillary )
                           ) (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

24. dcpdlogrequest (dcpdlogreq) The dcp request delog session command will allow the user to enter a delog session, and delog only dcp data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start stop time, delog by OI MTUL time, the OI MTUL start time, the OI MTUL stop time, the window number and a format id.

Example:

```

dcpdlogreq 2 log1 2 0 delog1 2 0 0 beg end -1 -1 -1 -1 -1 -1 ^
| | | | | | | | | | | | | | | |
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 - log file device      2 - log file name
(0 - floppy )           (this requires the user to enter the)
(1 - disk )             (name of the log file being delogged)
                          (up to 15 chars if the value of
                          (parameter 1 is 1. If it is 0 then )
                          (the value of this parameter is \0 )

3 - delog listing device 4 - output format
(0 - line printer)      (0 - 80 column )
(1 - CRT )              (1 - 132 column)
(2 - disk )

5 - delog file name      6 - delog format
(this requires the user to enter the)    (0 - hex dump )
(user to enter the name of the delog)    (1 - ascii dump )
(file up to 15 characters if the )        (2 - predefined format)
(device is disk, if it is not disk )      (3 - headers only )
(then the entry on the command string)    (4 - summary only )
(should be \0 )

```

```

7 - delog by time      8 - delog clock
  (0 - No )           (0 - GMT clock )
  (1 - Yes )          (1 - Alternate clock )

9 - start time        10 - stop time
  (format is ddd:hh:mm:ss.sss or )   (format is ddd:hh:mm:ss.sss )
  (the value is beg if not delogging) ( or the value end if not )
  (by time.)           (delogging by time )

11 - delog by OI MUTL time  12 - OI MUTL start time
  (1 - Yes )              (format same as parameter)
  (-1 - No )              (9, if the value of )
                           (parameter 11 is -1 then )
                           (the value here is -1 )

13 - OI MTUL stop time  14 - window number
  (format same as )       (values are 0-7 if the user wants a)
  (parameter 10, if )     (a specific window, if all windows )
  (the value of )         (are needed the value is -1 )
  (parameter 11 is )
  (-1 then the value)
  (here is -1 )

15 - format id
  (if the user wants a specific format)
  (the value here will be the format )
  (id, if all formats are required the)
  (value is -1 )

```

The health and status delog request command will allow the user to enter a health and status delog session, and delog only health and status data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process. This command will delog all message types which are classified as health and status. the message types are:

1. Initialization messages
2. Advisory messages
3. Error messages
4. General status messages
5. Interface messages
6. Operating System status messages
7. Queue status messages
8. Configuration messages
9. Acknowledgement messages
10. Restart messages
11. Verification messages
12. Test messages
13. Process messages

[illegible]

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

The health and status delog one request command will allow the user to enter a health and status delog session, and delog only health and status data type which are initialization messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

```
Example:
```

```
hsdlogreq 2 logl 2 0 delogl 2 0 0 beg end -1 -1 1 - log file device  
          |   |   |   |   |   |   |   |   |   |  
          1   2   3   4   5   6   7   8   9   10  11 12  
2 - log file name      3 - delog listing device  
  (this requires )     (0 - line printer)  
  (the user to )       (1 - CRT )  
  (enter the name)     (2 - disk )  
  (of the log file)  
  (being delogged )  
  (up to 15 chars )  
  (if the value of parameter 1 is 0)  
  (then the value here is \0)
```

- ```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

A-85

```

```

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



**Example:**

**A-87**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (nnd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

28. hs3dlogrequest (hs3dlogreq) The health and status delog two request command will allow the user to enter a health and status delog session, and delog only health and status data type which are error messages. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

Example:

```
hs3dlogreq 2 log1 ^ 2 0 0 delog1 ^ ^ ^ ^ -1 ^ 1 - log file device
| | | | | | | | | | | |
1 2 3 4 5 6 7 8 9 10 11 12
(0 - floppy )
(2 - disk )
```

- ```
2 - log file name
  (this requires )
  (the user to )
  (enter the name)
  (of the log file)
  (being delogged )
  (up to 15 chars )
  (if the value of parameter 1 is 0)
  (then the value here is \0 )

3 - delog listing device
  (0 - line printer)
  (1 - CRT )
  (2 - disk )

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )
```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

- for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start, stop time, originating application and originating process.

<b>hs3dlogreq</b>	<b>2</b>	<b>logl</b>	<b>2</b>	<b>0</b>	<b>deogl</b>	<b>2</b>	<b>0</b>	<b>beg</b>	<b>-l</b>	<b>-l</b>	<b>1 - log file device</b>
											(0 - floppy )
						•					(2 - disk )

- ```

    (if the value of parameter 1 is 0)
    (then the value here is \0 )

```

- A-91

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (nnd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

**A-93**

```

8 - - delog clock
    (0 - GMT clock )
    (1 - Alternate clock )

9 - - start time
    (format is ddd:hh:mm:ss.sss or )
    (the value is beg if not delogging)
    (by time.)

10 - - stop time
    (format is ddd:hh:mm:ss.sss or )
    (the value end if not delogging )
    (by time

11 - - originating application
    (support )
    (checkout )
    (ndd )
    (ancillary )
    (-1 all applications)

12 - - originating process
    (health stat )
    (user_if )
    (lan_if )
    (ancillary )
    (ndd )
    (data_generator )
    (scriptor )
    (score )
    (Delogger )

```



(hs6dlogreq)

**Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format                                7 - delog by time
(0 - hex dump )                                (0 - No )
(1 - ascii dump )                             (1 - Yes )
(2 - predefined format )
(3 - headers only )
(4 - summary only )

8 - delog clock                                9 - start time
(0 - GMT clock )                             (format is ddd:hh:mm:ss.sss or )
(1 - Alternate clock )                       (the value is beg if not delogging)
  (by time.)

10 - stop time                                11 - originating application
(format is ddd:hh:mm:ss.sss or )             (support )
(the value end if not delogging )             (checkout )
(by time )                                     (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
(health_stat )
(user_if )
(lan_if )
(ancillary )
(ndd )
(data_generator )
(scriptor )
(score )
(Delogger )

```

C-2

**Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

### Example:

A-99

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

A-101

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



**Example:**

```

4 - output format
  (0 - 80 column )
  (1 - 132 column)

5 - delog file name
  (this requires the user to enter the)
  (user to enter the name of the delog)
  (file up to 15 characters if the )
  (device is disk, if it is not disk )
  (then the entry on the command string)
  (should be \0 )

```

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



```

6 - delog format                                7 - delog by time
(0 - hex dump )                                (0 - No )
(1 - ascii dump )                              (1 - Yes )
(2 - predefined format )
(3 - headers only )
(4 - summary only )

8 - delog clock                                9 - start time
(0 - GMT clock )                               (format is ddd:hh:mm:ss.sss or )
(1 - Alternate clock )                         (the value is beg if not delogging)
  (by time.)

10 - stop time                                11 - originating application
(format is ddd:hh:mm:ss.sss or )              (support )
(the value end if not delogging )              (checkout )
(by time )                                     (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
(health_stat )
(user_if )
(lan_if )
(ancillary )
(nnd )
(data_generator )
(scriptor )
(score )
(Delgger )

```

**Example:**

A-107

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (ndd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (ndd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```

**Example:**

A-109

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

11 - originating application
  (support )
  (checkout )
  (nnd )
  (ancillary )
  (-1 all applications)

12 - originating process
  (health_stat )
  (user_if )
  (lan_if )
  (ancillary )
  (nnd )
  (data_generator )
  (scriptor )
  (score )
  (Delogger )

```



39. nasdlogrequest (nasdlogreq) The nascom delog session command will allow the user to enter a delog session, and delog only nascom data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start stop time, delog by OI MTUL time, the OI MTUL start time, the OI MTUL stop time, the specific MSID and which data stream.

Example:

```
nasdlogreq 2 logl 2 0 0 delogl 2 0 0 beg end -1 -1 -1 -1 -1 ^
| | | | | | | | | | | | | | |
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 - log file device          2 - log file name
(0 - floppy )              (this requires the user to enter the)
(1 - disk )                (name of the log file being delogged)
                             (up to 15 chars if the value of
                             (parameter 1 is 1. If it is 0 then )
                             (the value of this parameter is \0 )

3 - delog listing device    4 - output format
(0 - line printer)         (0 - 80 column )
(1 - CRT )                 (1 - 132 column)
(2 - disk )

5 - delog file name        6 - delog format
(this requires the user to enter the)    (0 - hex dump )
(user to enter the name of the delog)    (1 - ascii dump )
(file up to 15 characters if the )       (2 - predefined format)
(device is disk, if it is not disk )     (3 - headers only )
(then the entry on the command string)   (4 - summary only )
(should be \0 )
```

```

7 - delog by time      8 - delog clock
  (0 - No )           (0 - GMT clock )
  (1 - Yes )           (1 - Alternate clock )

9 - start time          10 - stop time
  (format is ddd:hh:mm:ss.sss or )   (format is ddd:hh:mm:ss.sss )
  (the value is beg if not delogging) ( or the value end if not )
  (by time.)              (delogging by time )

11 - delog by OI MUT1 time      12 - OI MUT1 start time
  (1 - Yes )                   (format same as parameter)
  (-1 - No )                   (9, if the value of )
                                (parameter 11 is -1 then )
                                (the value here is -1 )

13 - OI MUT1 stop time      14 - specific MSID
  (format same as )           (the 9-10 character MSID name is used)
  (parameter 10, if )         (make sure that the alpha characters )
  (the value of )             (are capitalized, if all MSID's are )
  (parameter 11 is )           (needed then use the value -1 )
  (-1 then the value)
  (here is -1 )

15 - data stream
  (If the user wants either stream 1 or)
  (stream 2 then the value will be 1 or)
  (2 . For both streams enter a -1 )

```

The OD frame delog session command will allow the user to enter a delog session, and delog only OD data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start stop time, delog by OI MTUL time, the OI MTUL start time, and the OI MTUL stop time.

**• All doses total to 200 mg / 2000 mg**

|           |           |               |           |           |                          |           |           |           |                    |                        |             |             |             |
|-----------|-----------|---------------|-----------|-----------|--------------------------|-----------|-----------|-----------|--------------------|------------------------|-------------|-------------|-------------|
| oddlogreg | $\hat{2}$ | $\text{logl}$ | $\hat{2}$ | $\hat{0}$ | $\widehat{\text{deogl}}$ | $\hat{2}$ | $\hat{0}$ | $\hat{0}$ | $\hat{\text{beg}}$ | $\widehat{\text{end}}$ | $- \hat{1}$ | $- \hat{1}$ | $- \hat{1}$ |
|           |           |               |           |           |                          |           |           |           |                    |                        |             |             |             |
|           | 1         | 2             | 3         | 4         | 5                        | 6         | 7         | 8         | 9                  | 10                     | 11          | 12          | 1           |

```

1 - log file device
  (0 - floppy )
  (1 - disk   )

2 - log file name
  (this requires the user to enter the)
  (name of the log file being delogged)
  (up to 15 chars if the value of )
  (parameter 1 is 1. If it is 0 then )
  (the value of this parameter is \0 )

```

```

3 - delog listing device
(0 - line printer)
(1 - CRT )
(2 - disk )

4 - output format
(0 - 80 column )
(1 - 132 column)

```

```

5 - delog file name
    (this requires the user to enter the )
    (user to enter the name of the delog)
    (file up to 15 characters if the )
    (device is disk, if it is not disk )
    (then the entry on the command string)
    (should be \0 )

6 - delog format
    (0 - hex dump )
    (1 - ascii dump )
    (2 - predefined format)
    (3 - headers only )
    (4 - summary only )

```

```

7 - delog by time
  (0 - No )
  (1 - Yes )
9 - start time
  (format is ddd:hh:mm:ss or )
  (the value is beg if not delogging)
  (by time.)
8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )
10 - stop time
  (format is ddd:hh:mm:ss.sss )
  ( or the value end if not )
  (delogging by time )
11 - delog by OI MUT1 time
  (1 - Yes )
  (-1 - No )
12 - OI MUT1 start time
  (format same as parameter)
  (9, if the value of )
  (parameter 11 is -1 then )
  (the value here is -1 )
13 - OI MTU1 stop time
  (format same as )
  (parameter 10, if )
  (the value of )
  (parameter 11 is )
  (-1 then the value)
  (here is -1 )

```

41. tdcpdlogrequest (tdcpdlogreq) The test dcp request delog session command will allow the user to enter a delog session, and delog only data generator dcp data type. He will be prompted for log file device, log file name, the delog listing device, the format if device is disk, the delog format, delog by time the delog clock, the start stop time, delog by OI MTUL time, the OI MTUL start time, the OI MTUL stop time, the window number and a format id.

**Example:**

```
tdcpdlogreq 2 log1 2 0 delog1 2 0 0 beg end -1 -1 -1 -1 -1 ^
      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
      1  2  3  4  5  6  7  8  9  10 11 12 13 14 15

1 - log file device      2 - log file name
(0 - floppy )           (this requires the user to enter the)
(1 - disk )             (name of the log file being delogged)
                        (up to 15 chars if the value of )
                        (parameter 1 is 1. If it is 0 then )
                        (the value of this parameter is \0 )

3 - delog listing device  4 - output format
(0 - line printer)       (0 - 80 column )
(1 - CRT )               (1 - 132 column)
(2 - disk )

5 - delog file name      6 - delog format
(this requires the user to enter the)   (0 - hex dump )
(user to enter the name of the delog)   (1 - ascii dump )
(file up to 15 characters if the )       (2 - predefined format)
(device is disk, if it is not disk )     (3 - headers only )
(then the entry on the command string)   (4 - summary only )
(should be \0 )
```

```

7 - delog by time      8 - delog clock
  (0 - No )           (0 - GMT clock )
  (1 - Yes )           (1 - Alternate clock )
9 - start time         10 - stop time
  (format is ddd:hh:mm:ss.sss or )   (format is ddd:hh:mm:ss.sss )
  (the value is beg if not delogging) ( or the value end if not )
  (by time.)           (delogging by time )

11 - delog by OI MUT1 time      12 - OI MUT1 start time
  (1 - Yes )                   (format same as parameter)
  (-1 - No )                   (9, if the value of )
                                (parameter 11 is -1 then )
                                (the value here is -1 )

13 - OI MTU1 stop time      14 - window number
  (format same as )           (values are 0-7 if the user wants a)
  (parameter 10, if )         (a specific window, if all windows )
  (the value of )             (are needed the value is -1 )
  (parameter 11 is )
  (-1 then the value)
  (here is -1 )

15 - format id
  (if the user wants a specific format)
  (the value here will be the format )
  (id, if all formats are required the)
  (value is -1 )

```

**Example:**

A-117

```

7 - delog by time      8 - delog clock
  (0 - No )           (0 - GMT clock )
  (1 - Yes )           (1 - Alternate clock )
9 - start time         10 - stop time
  (format is ddd:hh:mm:ss.sss or ) (format is ddd:hh:mm:ss.sss )
  (the value is beg if not delogging) ( or the value end if not )
  (by time.)           (delogging by time )

11 - delog by OI MUTl time 12 - OI MUTl start time
  (1 - Yes )           (format same as parameter)
  (-1 - No )           (9, if the value of )
                       (parameter 11 is -1 then )
                       (the value here is -1 )

13 - OI MUTl stop time 14 - specific MSID
  (format same as )    (the 9-10 character MSID name is used)
  (parameter 10, if )  (make sure that the alpha characters )
  (the value of )     (are capitalized, if all MSID's are )
  (parameter 11 is )   (needed then use the value -1 )
  (-1 then the value)
  (here is -1 )

15 - data stream
  (If the user wants either stream 1 or)
  (stream 2 then the value will be 1 or)
  (2 . For both streams enter a -1 )

```



43. toddlogrequest (toddlogreq) The test OD frame delog session command will allow the user to enter a delog session, and delog only data generator OD data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start stop time, delog by OI MTUL time, the OI MTUL start time, and the OI MTUL stop time.

Example:

```
toddlogreq 2 log1 2 0 0 delog1 2 0 0 beg end -1 -1 -1
| | | | | | | | | | | | | |
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 - log file device      2 - log file name
(0 - floppy )           (this requires the user to enter the)
(1 - disk )             (name of the log file being delogged)
                           (up to 15 chars if the value of
                           (parameter 1 is 1. If it is 0 then )
                           (the value of this parameter is \0 )

3 - delog listing device  4 - output format
(0 - line printer)       (0 - 80 column )
(1 - CRT )               (1 - 132 column)
(2 - disk )

5 - delog file name      6 - delog format
(this requires the user to enter the)   (0 - hex dump )
(user to enter the name of the delog)   (1 - ascii dump )
(file up to 15 characters if the )       (2 - predefined format)
(device is disk, if it is not disk )     (3 - headers only )
(then the entry on the command string)   (4 - summary only )
(should be \0 )
```

```

7 - delog by time          8 - delog clock
  (0 - No )                (0 - GMT clock )
  (1 - Yes )               (1 - Alternate clock )

9 - start time              10 - stop time
  (format is ddd:hh:mm:ss or ) (format is ddd:hh:mm:ss.sss )
  (the value is beg if not delogging) ( or the value end if not )
  (by time.)                (delogging by time )

11 - delog by OI MUTL time  12 - OI MUTL start time
  (1 - Yes )                (format same as parameter)
  (-1 - No )                (9, if the value of )
                             (parameter 11 is -1 then )
                             (the value here is -1 )

13 - OI MTUL stop time
  (format same as )
  (parameter 10, if )
  (the value of )
  (parameter 11 is )
  (-1 then the value)
  (here is -1 )

```

The user interface request delog session command will allow the user to enter a delog session, and delog only user interface data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start and stop time.

```

Example:
uifdlogreq 2 logl 2 0 delogl 2 0 0 beg end      1 - log file device
            | | | | | | | | | | | | | | | |    (0 - floppy )
            1 2 3 4 5 6 7 8 9 10                 (2 - disk  )

2 - log file name
   (this requires )
   (the user to )
   (enter the name)
   (of the log file)
   (being delogged )
   (up to 15 chars )
   (if the value of parameter 1 is 0)
   (then the value here is \0 )

3 - delog listing device
   (0 - line printer)
   (1 - CRT )
   (2 - disk )

```

A-121

```

6 - delog format
  (0 - hex dump )
  (1 - ascii dump )
  (2 - predefined format )
  (3 - headers only )
  (4 - summary only )

7 - delog by time
  (0 - No )
  (1 - Yes )

8 - delog clock
  (0 - GMT clock )
  (1 - Alternate clock )

9 - start time
  (format is ddd:hh:mm:ss.sss or )
  (the value is beg if not delogging)
  (by time.)

10 - stop time
  (format is ddd:hh:mm:ss.sss or )
  (the value end if not delogging )
  (by time )

```

45. utilldlogrequest (utilldlogreq) The utility request delog session command will allow the user to enter a delog session, and delog only HWC0 data type. He will be prompted for log file device, log file name, the delog listing device, the format for the delog, the delog file name if device is disk, the delog format, delog by time the delog clock, the start and stop time.

Example:

```
utilldlogreq 2 logl 2 0 delogl 2 0 0 beg end
      | | | | | | | | | |
      1 2 3 4 5 6 7 8 9 10
1 - log file device      2 - log file name
(0 - floppy )           (this requires the user to enter the)
(1 - disk )            (name of the log file being delogged)
                        (up to 15 chars if the value of )
                        (parameter 1 is 1. If it is 0 then )
                        (the value of this parameter is \0 )

3 - delog listing device 4 - output format
(0 - line printer)      (0 - 80 column )
(1 - CRT )              (1 - 132 column)
(2 - disk )

5 - delog file name      6 - delog format
(this requires the user to enter the)   (0 - hex dump )
(user to enter the name of the delog)   (1 - ascii dump )
(file up to 15 characters if the )       (2 - predefined format)
(device is disk, if it is not disk )     (3 - headers only )
(then the entry on the command string)   (4 - summary only )
(should be \0 )
```

- 7 - delog by time  
(0 - No )  
(1 - Yes )
- 8 - delog clock  
(0 - GMT clock )  
(1 - Alternate clock )
- 9 - start time  
(format is ddd:hh:mm:ss.sss or )  
(the value is beg if not delogging)  
(by time. )
- 10 - stop time  
(format is ddd:hh:mm:ss.sss )  
( or the value end if not )  
(delogging by time )

USER INTERFACE1. reqscrn (req)

The request screen command. This command would be used by a program and not used by a user.

Example:

req

2. relscrn (rel)

The release screen command. This command would be used by a program not a user.

Example:

rel

3. ackadv (acka)

The acknowledge pending advisory will show the next advisory in the advisory queue. This is the same function as depressing the F4 key.

Example:

acka

4. advh (ad)

The advisory history command will display all the advisory messages that have been acknowledged. The function is the same as depressing the F5 key and entering a return.

Example:

ad

### 5. cmdh (c)

The command history command will display the last 17 commands. The function is the same as depressing the F5 key and entering a "c".

#### Example:

c

### 6. menu (m)

The select menu command will prompt the user for the menu of his choice.

#### Example: m master

### 7. refresh (ref)

The refresh command will refresh the screen if system messages interrupt or if the user is switching from one workstation to another. The function is the same as depressing the F1 key.

#### Example:

ref

### 8. top (to)

The return to top command will return the user to the master menu (if the user is in the menu system) or it will go to the top of a report (if the user is viewing a report). The function is the same as depressing the F2 key.

#### Example:

to



### 9. history (hist)

The view history command will prompt the user for advisory or command history commands. This is the same function as depressing the F5 key and entering an "a" or a "c".

Example:

```
hist a      (a - advisory)
           (c - command )
```

### 10. view (v)

View a file command will prompt the user for the file name.

Example:

```
v /cas/reports/OVERALL
```

### 11. ackrprt (ackr)

Acknowledge pending report command will display on the screen a report that you requested. This is the same function as depressing the F3 key.

Example:

```
ackr
```

### 12. copy

The copy command will prompt the user for the file name to be copied and the name it is to be copied to. No example is given here since the information needed will exceed the 72 character limit on the command line.

### 13. delete (del)

The delete command will prompt the user for the file name of the file to be

deleted. No example is given here since the information needed will exceed the 72 character limit on the command line.

### 14. directory (dir)

The directory command will prompt the user for the directory the user wants a listing of.

Example:

dir /cas/support\_files

### 15. hexdump (hexd)

The hex dump command will dump a file in hex. It will prompt the user for the file name of the file to be dumped and the file name for the file which will hold the results.

Example:

hexd /cas/reports/dump1

### 16. ascdump (ascd)

The ASCII dump command will dump a file in ASCII. It will prompt the user for the file name of the file to be dumped and the file name for the file which will hold the results.

Example:

ascd /cas/reports/dump2

### 17. flpappend (flpa)

The append a file to a floppy will append a file to a floppy in the floppy drive.

Example:

flpa /cas/support\_files/MNO61ERA.d

### 18. flpformat (flpf)

The format floppy command will format a floppy in the floppy drive, this must be done before using a floppy to copy to. No other parameters are needed.

Example:

flpf

### 19. flplist (flpl)

The list floppy will list all the files on a users floppy which has been mounted in the floppy drive. No other parameters are needed.

Example:

flpl

### 20. flpread (flpr)

The floppy read command will read one or all files from the floppy to the hard disk. The user will be prompted for a file name or for all the files.

Example:

flpr /cas/support\_files/MNO61ERA.d to read a single file from the floppy.  
flpr /cas/support\_files to read multiple files under the same directory.

## 21. flpwrite (flpw)

The floppy write command will write a file to the floppy by overwriting what is on the floppy. This command should be used only for writing the first file on the floppy.

### Example:

```
flpw /cas/support_files/MNO61ERA.d to write a single file to the floppy.  
flpw /cas/support_files          to write multiple files from a directory to  
the floppy.
```

## 22. move

The move command will move a file from one directory to another. The user will be prompted for the file name of the file to be moved and the name of the destination file. No example is given since the information needed is greater than the 72 characters allowed on the command line.

## 23. print

The print command will send a file to the printer. The user will be prompted for the file name.

### Example:

```
print /cas/reports/OVERALL
```

## 24. processes (proc)

The active processes command will display the active processes running in a particular machine.

### Example:

```
proc 0
```

### 25. rcopy

The remote copy command will copy a file from one computer to the computer the user is on. The user will be prompted for the machine the file is located in and the name of the file, and the destination directory it is to be copied to. No example is given here since the information needed is greater than the 72 characters allowed on the command line.

### 26. rename (ren)

The rename command will rename a file on disk. The user will be prompted for the file name of the file to be renamed and its new name. The file name must include the path name where the file is located. No example is given since the information needed is greater than the 72 characters allowed on the command line.

### 27. rsend

The remote send command will send a file from the users's computer to another computer in the system. The user will be prompted for the file name of the file to be sent and then the machine it is to be sent and the name it is to be labeled. File name is not only the file name but its path name as well. No example is given since the information needed is greater than the 72 characters allowed on the command line

28. who

The who command will indicate who are logged on to the machine you are on.

Example:

who

29. ldcmds (ldc)

The load commands command will load a command set which is in the cmd\_sets directory. The user will be prompted for the name of the command file to be loaded. These files have a suffix of .cmd, this part of the name should not be given.

Example:

ldc log

30. line (li)

The adjust current report by lines will move the user through a report the user is viewing. The user will be prompted for the number of lines. A "+" and then a number will advance through the report for that number of lines. A "-" and a number will retreat through the report for that number of lines. A number without a "+" or "-" will move to that particular line.

Example:

li +5

### 31. page (pa)

The page command will move through a report at a page at a time. The user will be prompted for the number of pages. A "+" with a number advances through the report that number of pages. A "-" with a number will retreat through the report that number of pages.

Example:

pa +5

### 32. logon

The logon command will begin the process of logging onto the CAS system. The user will be prompted for the account he belongs to.

Example:

logon dfe

### 33. logoff

The logoff will log off the user. There are no other parameters.

Example:

logoff

### 34. user

The user command will indicate who is on the machine.

Example:

user

### 35. level

Example:

level

The level command will give the level of security the user has been assigned by his logon id and password. No other parameters are needed.

### 36. uifrestart (uifre)

Example:

uifre

The user interface restart command will restart the user interface process. No other parameters are needed.

### 37. sprint (spr)

Example:

spr 100 200

The selective print command will send a section of file to the printer. The user will be prompted for the file name of the file to be printed, the line at which the print is to start, and the line at which the print is to end.

### 38. rview (rv)

Example:

rv OVERALL

The report view command will view a report from the reports directory. The user will be prompted for the file name of the file to be viewed. The file name for this command is just the file name and not the whole path name as well.



### 39. rttylevel (rttyl)

The remote security level command will give the security of the remote terminal if one is active. There are no other parameters.

Example:

rttyl

### 40. rttyuser (rttyu)

The remote user command will indicate who the remote user is if one is active. There are no other parameters.

Example:

rttyu

### 41. perform (per)

The perform command is issued when a user wants to use a batch file (command file). The user will be prompted for the batch file name. All batch files have the extension .bat, but when using this command the extension is not given.

Example:

per logndcp

42. csacactive (csacact)

The CSAC active command directs user interface to send all commands to CSAC to be routed to the controlling workstation. No other parameters are necessary.

Example:

csacact

43. csacinactive (csacinact)

The CSAC inactive command directs user interface that all commands are to be routed locally. No other parameters are necessary.

Example:

csacinact

SUPERVISOR1. abort (ab)

The abort command causes the supervisor to terminate the specified process. Since the termination is not graceful, this command should only be used if the specified process is not responding to commands and all other attempts to halt the process have failed. Use of this command will cause abnormal termination advisories to be issued to the process. The abnormal termination code will be 011. process. The user will be prompted for the application and process.

## Example:

```
ab ndd ndd      1 Application (support, ndd, ancillary, or
  |             checkout)
  |             2 Process (heal_stat, ndd, ancillary,
  1             data_generator, scriptor,
                Delogger, or score)
```

2. haltproc (haltpr)

The halt process command causes the supervisor to send a halt command to the specified process. When a process receives a halt command it should perform a graceful shutdown of all of its functions followed by a normal termination. The user will be prompted for the application and process.

## Example:

```
haltpr ndd ndd  1 Application (support, ndd, ancillary, or
  |             checkout)
  |             2 Process (health_stat, ndd, ancillary,
  1             Delogger, scriptor, or score)
  2
```

### 3. restart (rest)

The restart command causes the supervisor to send a restart process command to the specified process. The restart command causes the process to gracefully halt what it is doing and to perform a warm restart. The user will be prompted for the application and the process.

#### Example:

```
rest ndd ndd      1 Application (support, ndd, ancillary or
                  |   | checkout)
                  |   |
                  1   2 Process (health_stat, ndd, ancillary,
                        Delogger, scriptor, or score)
```

### 4. exit

The exit command causes the supervisor to send a halt command to all active process in the system. The processes are given 20 seconds to halt. If they have not halted within that time period, the supervisor aborts the process. The user interface and health and status processes are halted after all other process have terminated. Once all processes have terminated, the supervisor cleans up the files used for inter-process communication, finally halting itself. \*\*note Any process which requires more than 20 seconds to gracefully halt (i.e. mission planning) should be halted first. Once these processes have halted the exit command may be safely used. No other parameters are needed.

#### Example:

```
exit
```

### 5. start (sta)

The start command will start any process. The user will be prompted for the application, process and 13 parameters. For unused parameters the word DONE should be inserted.

#### Example:

```
sta ndd ndd 51-e a rt ops
```

### 6. osdelay (osde)

The os delay command will set up the time interval in which the process os stats takes system information and outputs it in the HS MSG log file. The user will be prompted for the number of seconds this delay should be.

#### Example:

```
osde 20
```

CSAC1. csachalt (csach)

Example:

csach

The CSAC halt command will halt the CSAC process. There are no other parameters.

2. csacstart (csacst)

Example:

csacst

The CSAC start command will start the CSAC process. There are no other commands.

3. csac

Example:

csac

The CSAC menu command will call up the CSAC master menu. There are no other parameters.

4. local

Example:

local w/sl

local nddl

The local command will place any workstation in a local mode. This mode will not send messages to the CSAC workstation to be displayed on the CRT. The user will be prompted for the workstation to be put into local mode.

### 5. localall

Example:

localall

The local all command will put all workstations in a local mode. There are no other parameters.

### 6. remote

Example:

remote NDD1

The remote command will prompt the user for which NDD or workstation to put into the remote mode. The remote mode will have the particular workstation indicated send messages to the CSAC workstation. These messages will appear on the CSAC CRT.

### 7. remoteall

Example:

remoteall

The remote all command will have all workstations in the remote mode. There are no other parameters.

### 8. sendstatus (sendstat)

Example:

sendstat wsl

sendstat ndd9

sendstat all

The send status command will send a status message to the CSAC workstation. The user will be prompted for the workstation which will send its status.

9. datareport (data)

The data report command will produce a data stream report. The user will be prompted for the flight id and the data stream tag.

Example:

```
data 51-e rt      1 flight id
      |  |
      1  2      2 data stream tag (rt, sim, or pb)
```

10. nddreport (nddrep)

The NDD report command will produce an NDD report for the user. There are no other commands.

Example:

```
nddrep
```

11. networkreport (net)

The network report command will produce the Network Summary Report. There are no other parameters for this

Example:

```
net
```



## 12. wsreport (ws)

Example:

ws

The network report command will produce the Network Summary Report. There are no other parameters.

## 13. csacpause (csacp)

Example:

csacp

The CSAC pause command will pause all CSAC functions with the exception that CSAC will continue to read messages sent to it. However, no processing will be done to the messages. The csacpause command is useful when a time critical operation must be performed for a short period of time. While in the pause mode CASC will require considerable less cpu time to operate. CSAC will stay in the pause mode until until a csac resume or a csac halt command is received. All other commands will be ignored.

## 14. printcontrolws (printcon)

Example:

printcon

The print control workstation will print the name of the workstation under the control of CSAC. There are no other parameters for this command.

### 15. csacrestart (csacrest)

The CSAC restart command will restart the CSAC process, this command will be used if CSAC will not initialize. There are no other parameters for this command.

Example:

csacrest

### 16. csacresume (csacresu)

The CSAC resume command causes the CSAC process to leave the pause state and to resume processing all commands and status messages. There are no other parameters.

Example:

csacresu

### 17. controlws (control)

The control workstation command will give the user control over the workstation specified. The user will be prompted for the workstation to control.

Example:

control wsl

### 18. nddl

Example:

nddl

This is the control workstation command for NDDL. There are no other parameters.

30. ws4

Example:

ws4

This is the control workstation command for WS4. There are no other parameters.

31. ws5

Example:

ws5

This is the control workstation command for WS5. There are no other parameters.

32. ws6

Example:

ws6

This is the control workstation command for WS6. There are no other parameters.

33. ws7

Example:

ws7

This is the control workstation command for WS7. There are no other parameters.

34. ws8

Example:

ws8

This is the control workstation command for WS8. There are no other parameters.

35. ws9

This is the control workstation command for WS9. There are no other parameters.

Example:

ws8

36. setwstime

The set workstation time command will set the time on all the workstations that are on line. The user will be prompted for the new time.

Example:

setwstime 0216143585 (mmddhhmmyy)

37. systemreport (sys)

The system report command will generate a system status report. There are no other parameters for this command.

Example:

sys

38. applreport (applrep)

The application report command will prompt the user for the workstation the report is to be used for.

example:

applrep wsl

19. ddd2

Example:

ddd2

This is the control workstation command for NDD2. There are no other parameters.

20. ddd3

Example:

ddd3

This is the control workstation command for NDD3. There are no other parameters.

21. ddd4

Example:

ddd4

This is the control workstation command for NDD4. There are no other parameters.

22. ddd5

Example:

ddd5

This is the control workstation command for NDD5. There are no other parameters.

23. ddd6

Example:

ddd6

This is the control workstation command for NDD6. There are no other parameters.

24. ndd7

Example:

ndd7

This is the control workstation command for NDD7. There are no other parameters.

25. ndd8

Example:

ndd8

This is the control workstation command for NDD8. There are no other parameters.

26. ndd9

Example:

ndd8

This is the control workstation command for NDD9. There are no other parameters.

27. wsl

Example:

wsl

This is the control workstation command for WSL. There are no other parameters.

28. ws2

Example:

ws2

This is the control workstation command for WSL. There are no other parameters.

29. ws3

Example:

ws3

This is the control workstation command for WS3. There are no other parameters.

CHECKOUT SOFTWARE COMMANDS1. scrstart (scrsta)

The start scriptor command will start the interactive scriptor. The user will be prompted for the script name, the data base revision, and the data type.

## Example:

```
scrsta script1 61ERA dcp
      |      |      |
      1      2      3
```

1 - script file name  
(user determined name up to 15 chars)

2 - data base revision  
(this is composed of)  
(classic flight id )  
(61E, then the type )  
(R - realtime )  
(S - simulation )  
(P - playback )  
(and then the version)  
( A-Z )

3 - data type  
(the values are :  
(dcp  
(conf - for OD data  
(nascom  
(h&s - health & status  
(hwco - hardware checkout data)

2. scrstart\_hwco (scrstart\_hw) The start hardware checkout interactive scriptor will start the interactive scriptor for hardware checkout data. The user will be prompted for script file name.

## Example:

```
scrstart_hw script1
```

## 3. scrstart\_odf (scrstart\_o)

The start od interactive scriptor will start the interactive scriptor for OD data. The user will be prompted for the script file name and the data base version.

## Example:

```
scrstart_o od.script 6lERA
```

```
1 - script file name
(user determined name up to 15 chars)
```

```
1 1 1
```

```
2 - data base revision
(this is composed of)
(classic flight id )
(6lE, then the type )
(R - realtime )
(S - simulation )
(P - playback )
(and then the version)
( A-Z )
```



4. scrstart\_dcp (scrstart\_d)      The start dcp interactive scriptor will start the interactive scriptor for dcp data. The user will be prompted for the script file name and the data base version.

Example:

```
scrstart_d dcp.script 61ERA      1 - script file name
                                (user determined name up to 15 chars)
```

```
    |      |
    1      1
```

```
2 - data base revision
   (this is composed of)
   (classic flight id )
   (61E, then the type )
   (R - realtime      )
   (S - simulation    )
   (P - playback      )
   (and then the version)
   ( A-Z              )
```

## 5. scrstart\_n

The start nascom interactive scriptor will start the interactive scriptor for nascom data. The user will be prompted for the script file name and the data base version.

## Example:

```
scrstart_n nas.script 61ERA
```

```
  |      |      |
  1      1      1
```

```
1 - script file name
   (user determined name up to 15 chars)
```

```
2 - data base revision
   (this is composed of)
   (classic flight id )
   (61E, then the type )
   (R - realtime      )
   (S - simulation    )
   (P - playback      )
   (and then the version)
   ( A-Z              )
```

## 6. scrstart\_hs

The start health and status interactive script will start the interactive script for health and status messages. The user will be prompted for the script file name and the data base version.

## Example:

```
scrstart_hs hs.script 61ERA
```

```
  |      |
  1      1
```

```
1 - script file name
   (user determined name up to 15 chars)
```

```
2 - data base revision
   (this is composed of)
   (classic flight id )
   (61E, then the type )
   (R - realtime      )
   (S - simulation     )
   (P - playback       )
   (and then the version)
   ( A-Z               )
```

7. script (scri)

The append script line command will add script lines to the script you are working on when the interactive scriptor is running. The user will be prompted for the next script line he wants to add.

## Example:

```
scri form,oi=129,d11=20,d12=60
```

8. dgstart (dgsta)

The data generator start command will initialize the data generator for the desired data. The user will be prompted for the name of a log file (if he wants to route previously generated data), script file name, validate the script only, the output device the data is to be routed, the data type, the rate of output, and the NDD ID. This command must be followed by the go command for the data generator to start generating the desired data.

## Example:

```
dgsta null script1 no lan dcp 100 1
|      |      |      |      |      |
1      2      3      4      5      6      7

1 - previously generated data
  (this parameter with the value "null" indicates that data is to)
  (generated. If you have logged data and put the log file name )
  (in as the value of this parameter the data generator will use )
  (it to route this previously generated data. This log file )
  (must be in the /cas/application/checkout/log_files directory )

2 - script file name
  (user designated file name for the script which is to be used )
  (to generate the data. **note if the value for parameter 1 was )
  (the name of a log file - to route previously logged data - a )
  (script file name must still be given. The data generator needs)
  (it to determine an end time. )
```

```

3 - validate only
  (the value "yes" will indicate to the generator to only validate)
  (the script. If there are errors the first error will be brought)
  (to the screen and stop the process. If the value is "no" the data)
  (generator will generate data upon issuance of the go command
  )

4 -output device
  (the values are:
  (/dev/null - which will let the data generator run with no data)
  )
  (/dev/nciu - direct the output of the data generator to the NCIU)
  (
  driver
  )
  (lan - direct the output of the data generator to the LAN )
  (filename - direct the output of the data generator to a file )
  (
  **note this file must already exist
  )

5 - data type
  (hwco - hardware checkout data )
  (dcp
  )
  (conf - OD data
  )
  (nascom
  )
  (h&s - health & status messages)

6 - rate of output
  (values range from 1-200, this number represents the percentage)
  (of output rate, ex a 5 would be 5% of normal output rate )

7 - NDD ID
  ( values are 1-9, these must be used even if the data type is )
  (not dcp
  )

```

## 9. dgstart\_odef (dgstart\_o)

The data generator od start command will initialize the data generator for OD data. The user will be prompted for the name of a log file (if he wants to route previously generated data), script file name, validate the script only, the output device the data is to be routed, the rate of output, and the NDD ID. This command must be followed by the go command for the data generator to start generating the desired data.

## Example:

```
dgstart_o null script1 no /dev/nciu 100 1
```

```

  |      |      |      |      |
  1      2      3      4      5      6

```

## 1 - previously generated data

(this parameter with the value "null" indicates that data is to) (generated. If you have logged data and put the log file name) (in as the value of this parameter the data generator will use) (it to route this previously generated data. This log file) (must be in the /cas/application/checkout/log\_files directory)

## 2 - script file name

(user designated file name for the script which is to be used) (to generate the data. \*\*note if the value for parameter 1 was) (the name of a log file - to route previously logged data - a) (script file name must still be given. The data generator needs) (it to determine an end time.)

## 3 - validate only

(the value "yes" will indicate to the generator to only validate) (the script. If there are errors the first error will be brought) (to the screen and stop the process. If the value is "no" the data) (generator will generate data upon issuance of the go command)

```
4 -output device
  (the values are:
  (/dev/null - which will let the data generator run with no data)
  (
  (/dev/nciu - direct the output of the data generator to the NCIU)
  (
  (lan - direct the output of the data generator to the LAN)
  (filename - direct the output of the data generator to a file)
  (
  (**note this file must already exist
  )
  )
  )
5 - rate of output
  (values range from 1-200, this number represents the percentage)
  (of output rate, ex a 5 would be 5% of normal output rate
  )
6 - NDD ID
  ( values are 1-9, these must be used even if the data type is )
  (not dcp
  )
```

## 10. dgstart\_dcp (dgstart\_d)

The data generator dcp start command will initialize the data generator for DCP data. The user will be prompted for the name of a log file (if he wants to route previously generated data), script file name, validate the script only, the output device the data is to be routed, the rate of output, and the NDD ID. This command must be followed by the go command for the data generator to start generating the desired data.

## Example:

```
dgstart_d null script1 no lan 100 1
```

```
  |   |   |   |   |   |
  1   2   3   4   5   6
```

## 1 - previously generated data

(this parameter with the value "null" indicates that data is to be generated. If you have logged data and put the log file name) (in as the value of this parameter the data generator will use) (it to route this previously generated data. This log file) (must be in the /cas/application/checkout/log\_files directory)

## 2 - script file name

(user designated file name for the script which is to be used) (to generate the data. \*\*note if the value for parameter 1 was) (the name of a log file - to route previously logged data - a) (script file name must still be given. The data generator needs) (it to determine an end time.)

## 3 - validate only

(the value "yes" will indicate to the generator to only validate) (the script. If there are errors the first error will be brought) (to the screen and stop the process. If the value is "no" the data) (generator will generate data upon issuance of the go command)



- 4 -output device  
 (the values are:  
 (/dev/null - which will let the data generator run with no data)  
 (            - being output.  
 (/dev/nciu - direct the output of the data generator to the NCIU)  
 (            - driver  
 (lan        - direct the output of the data generator to the LAN )  
 (filename   - direct the output of the data generator to a file )  
 (            \*\*note this file must already exist  
              )  
 )
- 5 - rate of output  
 (values range from 1-200, this number represents the percentage)  
 (of output rate, ex a 5 would be 5% of normal output rate )  
 )
- 6 - NDD ID  
 ( values are 1-9, these must be used even if the data type is )  
 (not dcp  
              )  
 )

## 11. dgstart\_hs (dgstart\_h)

The data generator health and status start command will initialize the data generator for health and status messages. The user will be prompted for the name of a log file (if he wants to route previously generated data), script file name, validate the script only, the output device the data is to be routed, the rate of output, and the NDD ID. This command must be followed

by the go command for the data generator to start generating the desired data.

## Example:

```
dgstart_h ^ null script1 no lan 100 1
```

```
  |      |      |      |      |
  1      2      3      4      5      6
```

## 1 - previously generated data

(this parameter with the value "null" indicates that data is to be generated. If you have logged data and put the log file name) (in as the value of this parameter the data generator will use) (it to route this previously generated data. This log file) (must be in the /cas/application/checkout/log\_files directory)

## 2 - script file name

(user designated file name for the script which is to be used) (to generate the data. \*\*note if the value for parameter 1 was) (the name of a log file - to route previously logged data - a) (script file name must still be given. The data generator needs) (it to determine an end time.)

## 3 - validate only

(the value "yes" will indicate to the generator to only validate) (the script. If there are errors the first error will be brought) (to the screen and stop the process. If the value is "no" the data) (generator will generate data upon issuance of the go command)

- 4 -output device  
(the values are:  
(/dev/null - which will let the data generator run with no data)  
(            - being output.  
(/dev/nciu - direct the output of the data generator to the NCIU)  
(            - driver  
(lan        - direct the output of the data generator to the LAN )  
(filename - direct the output of the data generator to a file )  
(            \*\*note this file must already exist  
)
- 5 - rate of output  
(values range from 1-200, this number represents the percentage)  
(of output rate, ex a 5 would be 5% of normal output rate )
- 6 - NDD ID  
( values are 1-9, these must be used even if the data type is )  
(not dcp )

## 12. dgstart\_n

The data generator nascom start command will initialize the data generator for NASCOM block data. The user will be prompted for the name of a log file (if he wants to route previously generated data), script file name, validate the script only, the output device the data is to be routed,

the rate of output, and the NDD ID. This command must be followed by the go command for the data generator to start generating the desired data.

## Example:

```

dgstart_n null script1 no lan 100 1
      |      |      |      |      |
      1      2      3      4      5      6
1 - previously generated data
   (this parameter with the value "null" indicates that data is to)
   (generated. If you have logged data and put the log file name )
   (in as the value of this parameter the data generator will use )
   (it to route this previously generated data. This log file )
   (must be in the /cas/application/checkout/log_files directory )

2 - script file name
   (user designated file name for the script which is to be used )
   (to generate the data. **note if the value for parameter 1 was )
   (the name of a log file - to route previously logged data - a )
   (script file name must still be given. The data generator needs)
   (it to determine an end time.)

```

- 3 - validate only  
 (the value "yes" will indicate to the generator to only validate)  
 (the script. If there are errors the first error will be brought)  
 (to the screen and stop the process. If the value is "no" the data)  
 (generator will generate data upon issuance of the go command )
- 4 -output device  
 (the values are:  
 (/dev/null - which will let the data generator run with no data)  
 ( being output.  
 (/dev/nciu - direct the output of the data generator to the NCIU)  
 ( driver  
 (lan - direct the output of the data generator to the LAN )  
 (filename - direct the output of the data generator to a file )  
 ( \*\*note this file must already exist )
- 5 - rate of output  
 (values range from 1-200, this number represents the percentage)  
 (of output rate, ex a 5 would be 5% of normal output rate )
- 6 - NDD ID  
 ( values are 1-9, these must be used even if the data type is )  
 (not dcp )

### 13. dggg (dgg)

The data generator command will start the initialized data generator to output the desired data. The user will be prompted for the process name.

#### Example:

dgg data\_generator The process name data generator is used if only one data generator is active. If a second data generator is active its process name is data\_generato2 etc.

#### 14. scorstart (scorst)

The scoring start command will initialize the scoring process. The user will be prompted for data type, the script file name, the input device or file name, the advisories to be suppressed, the output list device or file, the start time, the stop time, the scoring mode, hbr/lbr rate, and the NDD ID. After initialization is complete the score go command must be issued for the scoring process to begin.

#### Example:

```
scorst dcp score.scrost dcp.log 0 score.list1 1 20 a hbr 1
|      |      |      |      |      |      |      |
1      2      3      4      5      6      7      8      9      10
```

- 1 - data type  
 ( valid entries are:  
 ( dcp - data cycle packets )  
 ( hwco - hardware checkout data pattern)  
 ( nascom - NASCOM blocks )
- 2 - script file name  
 (the user input script file to which the data in the log file )  
 (is to be compared to. This file is a prediction of what the )  
 (the data will look like. The file name can be up to 15 )  
 (characters and follows the UNIX naming convention. )
- 3 - input device or file  
 (the valid values will either be "/dev/nciu" when scoring hwco )  
 (data or the log file name when scoring DCP or NASCOM data. The)  
 (log file name can be up to 15 characters and follows the UNIX )  
 (naming convention. )

- 4 - advisories to be suppressed  
(valid values are the numbers which correspond to particular )  
(error conditions. For further information on the error )  
(conditions consult the design specification. More than one )  
(number can be used, they must be separated by a space. )
- 5 - output list device or file  
(the valid value here is the name of the file the user wants )  
(the error reporting to be routed to. This file will be created)  
(in the /cas/application/checkout/score\_files directory. )
- 6 - start time  
(the valid entry is the time in seconds you want to begin the )  
(comparison from the script file and log file you designated )  
(earlier. )
- 7 - end time  
(the valid entry is the time in seconds you want to end the )  
(comparison from the script file and log file you designated )  
(earlier. )

- 8 - scoring mode  
 (the valid entries are:  
 (s - single error mode. Upon the detection of the first error )  
 ( the scoring process stops. )  
 (a - all error mode. The scoring process will report all errors)  
 ( and only halt upon completion of the end time. )  
 (l - log only mode, after initializing the scoring process in )  
 ( this mode you must give the loge enable command - loge - )
- 9 - lbr/hbr (low bit rate/high bit rate)  
 (this parameter is only applicable for Hwco data. It must, )  
 (however, be present even if the data type is not hwco. The )  
 (valid entries are "lbr" for low bit rate or "hbr" for high bit )  
 (rate. )
- 10 - NDD ID  
 (this parameter is only applicable for DCP data. It must, )  
 (however, be present even if the data type is not dcp. The )  
 (valid entries are 1-4 and 9. )



15. scorstart\_hwco (scorstart\_h) The score start HWCO command will initialize the scoring process for hwco data. The user will be prompted for the script file name, the input device, the advisories to be suppressed, the output list device or file, the start tie, the end time, the scoring mode, and if the data is lbr or hbr. After initialization is complete the score go command must be issued to commence the scoring process.

Example:

```
scorstart_h score_ script dcp.log 0 score_ list1 1 20 a hbr
1      |      |      |      |      |      |      |
      1      2      3      4      5      6      7      8
```

- 1 - script file name  
(the user input script file to which the data in the log file )  
(is to be compared to. This file is a prediction of what the )  
(the data will look like. The file name can be up to 15 )  
(characters and follows the UNIX naming convention. )
- 2 - input device or file  
(the valid values will either be "/dev/nciu" when scoring hwco )  
(data or the log file name when scoring DCP or NASCOM data. The)  
(log file name can be up to 15 characters and follows the UNIX )  
(naming convention. )
- 3 - advisories to be suppressed  
(valid values are the numbers which correspond to particular )  
(error conditions. For further information on the error )  
(conditions consult the design specification. More than one )  
(number can be used, they must be separated by a space. )

- 4 - output list device or file  
 (the valid value here is the name of the file the user wants )  
 (the error reporting to be routed to. This file will be created)  
 (in the /cas/application/checkout/score\_files directory. )
- 5 - start time  
 (the valid entry is the time in seconds you want to begin the )  
 (comparison from the script file and log file you designated )  
 (earlier. )
- 6 - end time  
 (the valid entry is the time in seconds you want to end the )  
 (comparison from the script file and log file you designated )  
 (earlier. )
- 7 - scoring mode  
 (the valid entries are:  
 (s - single error mode. Upon the detection of the first error )  
 ( the scoring process stops. )  
 (a - all error mode. The scoring process will report all errors)  
 ( and only halt upon completion of the end time. )  
 (l - log only mode, after initializing the scoring process in )  
 ( this mode you must give the loge enable command - loge - )
- 8 - lbr/hbr (low bit rate/high bit rate)  
 (this parameter is only applicable for HWC0 data. It must, )  
 (however, be present even if the data type is not hwco. The )  
 (valid entries are "lbr" for low bit rate or "hbr" for high bit)

16. hwcostart (hwcost)

The HWCO start command will initialize the scoring process for hwco data. The user will be prompted for the script file name, the input device, the advisories to be suppressed, the output list device or file, the start tie, the end time, the scoring mode, and if the data is lbr or hbr. This command is similar to the start HWCO command, the difference is that if the user so desires to use the default values for the prompts he will be defaulting for the following information:

```
date type - hwco
script file name - hwco.script
input device or file - /dev/nciu
advisories to be suppressed - 0
output list device or file - hwco.list
start time - 1
end time - 10
scoring mode - a
lbr/hbr - hbr
```

After initialization is complete the score go command must be issued to commence the scoring process.

## Example:

```
hwcost hwco score.script dcp.log 0 score.list 1 20 a hbr
1      |      |      |      |      |      |      |
      1      2      3      4      5      6      7      8      9
```

```
1 - data type
   ( valid entries are:
     ( hwco - hardware checkout data pattern)
```

- 2 - script file name  
 (the user input script file to which the data in the log file )  
 (is to be compared to. This file is a prediction of what the )  
 (the data will look like. The file name can be up to 15 )  
 (characters and follows the UNIX naming convention. )
- 3 - input device or file  
 (the valid values will either be "/dev/nciu" when scoring hwco )  
 (data or the log file name when scoring DCP or NASCOM data. The)  
 (log file name can be up to 15 characters and follows the UNIX )  
 (naming convention. )
- 4 - advisories to be suppressed  
 (valid values are the numbers which correspond to particular )  
 (error conditions. For further information on the error )  
 (conditions consult the design specification. More than one )  
 (number can be used, they must be separated by a space. )
- 5 - output list device or file  
 (the valid value here is the name of the file the user wants )  
 (the error reporting to be routed to. This file will be created)  
 (in the /cas/application/checkout/score\_files directory. )
- 6 - start time  
 (the valid entry is the time in seconds you want to begin the )  
 (comparison from the script file and log file you designated )  
 (earlier. )

- 7 - end time  
 (the valid entry is the time in seconds you want to end the )  
 (comparison from the script file and log file you designated )  
 (earlier. )
- 8 - scoring mode  
 (the valid entries are:  
 (s - single error mode. Upon the detection of the first error )  
 ( the scoring process stops. )  
 (a - all error mode. The scoring process will report all errors)  
 ( and only halt upon completion of the end time. )  
 (l - log only mode, after initializing the scoring process in )  
 ( this mode you must give the log enable command - loge - )
- 9 - lbr/hbr (low bit rate/high bit rate)  
 (this parameter is only applicable for HWC0 data. It must, )  
 (however, be present even if the data type is not hwco. The )  
 (valid entries are "lbr" for low bit rate or "hbr" for high bit )  
 (rate. )

17. scorstart\_dcp (scorstart\_d) The score start DCP command will initialize the scoring process for DCP data. The user will be prompted for the script file name, the input device, the advisories to be suppressed, the output list device or file, the start tie, the end time, and the NDD ID. After initialization is complete the score go command must be issued to commence the scoring process.

Example:

```
scorstart_d score_script dcp_log 0 score_list 1 20 2
|      |      |      |      |      |
1      2      3      4      5      6      7
```

- 1 - script file name  
 (the user input script file to which the data in the log file )  
 (is to be compared to. This file is a prediction of what the )  
 (the data will look like. The file name can be up to 15 )  
 (characters and follows the UNIX naming convention. )
- 2 - input device or file  
 (the valid values will either be "/dev/nciu" when scoring hwco )  
 (data or the log file name when scoring DCP or NASCOM data. The)  
 (log file name can be up to 15 characters and follows the UNIX )  
 (naming convention. )
- 3 - advisories to be suppressed  
 (valid values are the numbers which correspond to particular )  
 (error conditions. For further information on the error )  
 (conditions consult the design specification. More than one )  
 (number can be used, they must be separated by a space. )

- 4 - output list device or file  
(the valid value here is the name of the file the user wants )  
(the error reporting to be routed to. This file will be created)  
(in the /cas/application/checkout/score\_files directory. )
- 5 - start time  
(the valid entry is the time in seconds you want to begin the )  
(comparison from the script file and log file you designated )  
(earlier. )
- 6 - end time  
(the valid entry is the time in seconds you want to end the )  
(comparison from the script file and log file you designated )  
(earlier. )
- 7 - NDD ID  
(the valid entries are 1-4 and 9 )

## 18. scorstart\_n

The score start NASCOM command will initialize the scoring process for NASCOM data. The user will be prompted for the script file name, the input device, the advisories to be suppressed, the output list device or file, the start tie, the end time, and the NDD ID. After initialization is complete the score go command must be issued to commence the scoring process.

## Example:

```

scorstart_n score_ script dcp,log 0 score_list 1 20 2
      |      |      |      |      |      |
      1      2      3      4      5      6 7

```

- 1 - script file name  
 (the user input script file to which the data in the log file )  
 (is to be compared to. This file is a prediction of what the )  
 (the data will look like. The file name can be up to 15 )  
 (characters and follows the UNIX naming convention. )
- 2 - input device or file  
 (the valid values will either be "/dev/nciu" when scoring hwco )  
 (data or the log file name when scoring DCP or NASCOM data. The)  
 (log file name can be up to 15 characters and follows the UNIX )  
 (naming convention. )
- 3 - advisories to be suppressed  
 (valid values are the numbers which correspond to particular )  
 (error conditions. For further information on the error )  
 (conditions consult the design specification. More than one )  
 (number can be used, they must be separated by a space. )



- 4 - output list device or file  
(the valid value here is the name of the file the user wants )  
(the error reporting to be routed to. This file will be created)  
(in the /cas/application/checkout/score\_files directory. )
- 5 - start time  
(the valid entry is the time in seconds you want to begin the )  
(comparison from the script file and log file you designated )  
(earlier. )
- 6 - end time  
(the valid entry is the time in seconds you want to end the )  
(comparison from the script file and log file you designated )  
(earlier. )
- 7 - NDD ID  
(the valid entries are 1-4 and 9 )

### 19. scogo (scog)

The score go command will start the scoring process after it has been initialized. There are no other parameters.

Example:

scog

### 20. scorview (scorv)

The view scoring file will allow the user to view the default list file (cosw\_results) on the screen. This file is in the directory /cas/application/checkout/score\_files. This file is produced if the user has taken the default file name for the list file in the scoring process. There are no other parameters.

Example:

scorv

### 21. scriptfiles (scriptfi)

The show script files command will list all the script files in the /cas/application/checkout/script\_files directory. There are no other parameters for this command.

Example:

scriptfi

## 22. stopscr

The halt scriptor process will halt the interactive scriptor. There are not other parameters necessary.

Example:

stopscr

## 23. stopdq (stopd)

The halt data generator command will stop the data generator process. No other parameters are necessary.

Example:

stopd

## 24. stopsco

The halt scoring process command will stop the scoring process. No other parameters are necessary.

HEALTH and STATUS1. hrestart (hres)

The H&S restart command will restart the health and status process. When this command is given it will re-set the H&S table of processes. No other parameters are needed.

Example:

hres

2. pause (pas)

The H&S pause command will pause the H&S process. It will not process any messages to the screen. No other parameters are needed.

Example:

pau

3. hresume (hresu)

The H&S resume command will resume the H&S processing of messages to the screen. This command is issued after the pause command. No other parameters are needed.

Example:

hresu

4. hsdelete (hsdel)

The H&S delete command will

5. mutdisable (mutdis)

The H&S disable mute command will

6. mutenable (muten)

The H&S enable mute command will

7. hsreinitialize (hsreinit)

The H&S reinitialization command will reinitialize the health and status process. When this is done the health and status table

8. happreport (happ)

The H&S application report command will prompt the user for the application he wants a report on.

Example:

happ ancillary

9. hlogreport (hlog)

The H&S logging report command will bring up the report on the types of logging currently being done in the system. No other parameters are needed. When the report is ready for viewing, the F3 area on the screen will be illuminated. Pressing the F3 button brings it to the screen for viewing.

Example:

hlog

10. hifreport (hif)

The H&S interface report command will bring up the report on all the active interfaces at the current time. No other parameters are needed. When the report is ready the F3 area on the screen will be illuminated. Pressing the F3 button brings it to the screen for viewing.

Example:

hif

### 11. hovrreport (hovr)

The H&S overall report command will bring up the report of all the currently active applications. No other parameters are needed. When the report is ready the F3 area on the screen will be illuminated. Pressing the F3 button bring it to the screen for viewing.

#### Example:

hovr

### 12. hcsacstatus (hcsacstat)

The send CSAC status command will activate H&S to send CSAC a status message. No other parameters are needed.

#### Example:

hcsacstat

### MISSION PLANNING

#### 1. haltmissionplan (haltm)

The halt mission planning command will halt the mission planning application. No other parameters are necessary.

#### Example:

haltm

#### 2. sendm

The send status to H&S command will activate the mission planning application to send status to Health and Status. No other parameters are necessary.

Example:

sendm

3. startmissionplan (startm)

The start mission planning command will  
start the mission planning application.  
The user will be prompted his mission  
planning ID.

Example:

startm 200

DEBUG COMMANDS\*\*\*\*\* CSAC DEBUG COMMANDS \*\*\*\*\*1. csacdebug (csacd)

The start CSAC in debug mode initiates the CSAC process and its associated processes. This command allows up to nine debug flags to be turned on prior to the start of CSAC initialization. After entering the csac debug mode command, the user will be prompted for the nine debug flags. If there are less than nine flags to set prior to initialization, just press the carriage return and the value "DONE" will be substituted for a debug flag number. When CSAC is operating, an advisory stating that the CSAC initialization is complete, is displayed on the terminal. The health and status and user interface processes on the CSAC workstation are placed into the remote control mode. In the remote control mode, the health and status process will ship a copy of each advisory to the CSAC process. The user interface process will send all commands (with the exception of user interface commands) to the CSAC process. The CSAC process will send all non CSAC commands to the workstation or ndd which is the current control workstation. The control workstation defaults at initialization time to the workstation CSAC is running on.



csacd 0 2 DONE DONE DONE DONE DONE

**DEBUG FLAGS** (modules alphabetically listed)

A-183

|    |              |                                                          |
|----|--------------|----------------------------------------------------------|
| 63 | xndd_disp.c  | formats display segment for one ndd                      |
| 64 | xndd_rept.c  | generates an ndd summary report                          |
| 34 | xnet_rept.c  | generates a network summary report                       |
| 35 | xnext_cmd.c  | removes the next command from the queue                  |
| 36 | xnext_msg.c  | removes the next message from the pipe                   |
| 37 | xpproc_msg.c | performs CSAC message preprocessing                      |
| 38 | xproc_ack.c  | processes an acknowledgement message                     |
| 39 | xproc_adv.c  | processes an advisory message                            |
| 40 | xproc_init.c | processes an initialization message                      |
| 41 | xproc_rept.c | processes a report message                               |
| 42 | xproc_rst.c  | processes a restart message                              |
| 43 | xproc_stat.c | processes a status message                               |
| 10 | xqueue_msg.c | sends a queue to the proper process                      |
| 44 | xremote.c    | sends a go to remote command                             |
| 45 | xrestart.c   | processes a restart command                              |
| 46 | xsend_cmd.c  | sends a command to a remote workstation                  |
| 69 | xsend_stat.c | sends a send status command                              |
| 47 | xserv_cmd.c  | services CSAC commands                                   |
| 48 | xserv_msg.c  | services CSAC messages                                   |
| 49 | xspecial.c   | decodes CSAC special commands                            |
| 50 | xsrch_tab.c  | searches w/s tables for a w/s name                       |
| 2  | xstart_lan.c | starts the csac_lan_if process                           |
| 51 | xstat_cmd.c  | sends the send_status command                            |
| 52 | xstat_rept.c | generates a workstation status report                    |
| 53 | xstatus.c    | polls workstations for their status                      |
| 58 | xtime.c      | sends a command to set time and date in all workstations |
| 54 | xsys_rept.c  | generates a CAS system report                            |
| 10 | xunix_err.c  | process UNIX errors                                      |
| 55 | xuser_cmd.c  | decodes CSAC user (report) commands                      |
| 56 | xvalid_msg.c | validates CSAC messages                                  |
| 61 | xverif_cmd.c | verifies initialization messages                         |
| 57 | xwho.c       | sends requests for init messages                         |
| 65 | xws_disp.c   | formats a w/s summary display segment                    |
| 66 | xws_rept.c   | generates a w/s summary report                           |
| 67 | xws_rmt.c    | places UIF and H&S into the remote mode                  |
| 59 | xxmmt_cmd.c  | transmits a command via the LAN                          |

## 2. cdbug (cdb)

The debug CSAC command causes the CSAC process to set or reset one of the CSAC debug flags. When a debug flag is turned on, it causes several advisory messages to be sent to the user interface for display. The advisory messages are useful in tracing the flow through the program logic. Most of the debug flags apply only to one module in the CSAC process. With the exception of modules which are frequently executed, most of the modules provide advisories when they are entered and exited. These advisories are omitted in frequently called modules since they tend to erase useful information from the viewing area of the screen. For an example and a listing of flags see csacdebug command.

## 3. cmdreport (cmdr)

The command report command causes CSAC to generate a report which lists the commands sent to remote workstations which have not been acknowledged. No other parameters are needed.

Example:

cmdr

## Example of report

## Outstanding Commands to Remote Workstations

| Msg Num | Destination  | Cmd Token | Time Sent    |
|---------|--------------|-----------|--------------|
| 23      | ndd1 support | 830       | 218:14:37:25 |
| 27      | ndd1 ndd     | 14        | 218:14:37:31 |

4. statusreport (statusrep)

The status report command causes CSAC to generate a report containing all of the information in the workstation table record for the specified workstation. The user will be prompted for the name of the workstation (w/s or ndd) needed.

## Example:

statusrep ndd2

## Example of a report:

## CSAC Workstation Status record for ndd2

```

name: ndd2                      health: OK
init time: 218:14:35:23
last time: 218:14:37:28
number of applications: 2
user mode: REMOTE                pause mode: OFF
nominal errors = 0                urgent errors = 0
flight 1: 51-E                    stream: RT
flight 2:                          stream:
message counter = 25              msg time err ctr = 0
current msg time: 218:14:37:27
last status recv: 218:14:37:28
solicited mag time: 218:14:37:27
load averages: 2.390000          2.190000          1.580000
cpu idle: 0                      cpu user: 0          cpu syst: 0
                                quality: OK
                                quality:

```

## 5. csacack

The csac acknowledge command causes CSAC to send a test acknowledgement message to itself from the specified workstation. The message is written by CSAC onto the CSAC message pipe. This command provides a way to test CSAC's response to acknowledgement messages from different workstation's and ndd's. The user will be prompted for the originating workstation, the message number and the message transfer status.

Example:

```
csacack ndd9 10000 1
      |      |      |
      1      2      3
```

- 1 - originating workstation  
(the name of the originating workstation must be in the list of CAS)  
(nodes found in the file /cas/application/csac/NET\_NAMES. The use )  
(of any other name may cause the CSAC process to crash. Presently )  
( the node names are: wsl, ws2, nddl, ndd2, ndd3, ndd4 and ndd9. )
- 2 - message number  
(valid range is 0-32000)
- 3 - transfer status  
(The possible codes returned from the enqueue module are: )  
( 0 - transfer ok )  
( 1 - an error occurred during the transfer )  
( 3 - no queue has been defined for the destination process )  
( 4 - the destination queue is full )

## 6. csacadv

The csac advisory command causes CASC to send a test advisory message to itself from the specified workstation. The message is written by CSAC onto the CSAC message pipe. This command provides a way to test CSAC's response to advisory messages from different workstation's and ndd's. The user will be prompted for the originating workstation (workstation or ndd).

### Example:

```
csacadv wsl
      |
      | originating workstation
      | (the name of the originating workstation must be in the list of CAS)
      | (nodes found in the file /cas/application/csac/NET_NAMES. The use )
      | (of any other name may cause the CSAC process to crash. Presently )
      | ( the node names are: wsl, ws2, nddl, ndd2, ndd3, ndd4 and ndd9. )
```

## 7. csacinit

The csac initialization command causes CSAC to send a test initialization message to itself from the specified workstation. The message is written by CSAC onto the CSAC message pipe. This command provides a way to test CSAC's response to initialization messages from different workstation's and ndd's. The user will be prompted for the originating workstation and the maximum message interval.

## Example:

```
csacinit ndd3 15
      ^
      | |
      1 2
```

- 1 - originating workstation  
 (the name of the originating workstation must be in the list of CAS)  
 (nodes found in the file /cas/application/csac/NET\_NAMES. The use )  
 (of any other name may cause the CSAC process to crash. Presently )  
 ( the node names are: wsl, ws2, nddl, ndd2, ndd3, ndd4 and ndd9. )
- 2 - maximum message interval  
 (The time in seconds which must elapse before CSAC automatically )  
 (polls for status. CSAC will poll for status every ten seconds )  
 (thereafter until a status response is received from the )  
 (workstation or ndd. )

8. csacrestart (csacres)

The send a CSAC restart message command will

## 9. csactermminate (csacterm)

The csac terminate command causes CASC to send a test termination message to itself from the specified workstation. The message is written by CSAC onto the CSAC message pipe. This command provides a way to test CSAC's response to termination messages from different workstation's and ndd's. The user will be prompted for the originating workstation (workstation or ndd).

### Example:

```
csacterm wsl
```

```
|
  originating workstation
  (the name of the originating workstation must be in the list of CAS)
  (nodes found in the file /cas/application/csac/NET_NAMES. The use )
  (of any other name may cause the CSAC process to crash. Presently )
  ( the node names are: wsl, ws2, nddl, ndd2, ndd3, ndd4 and ndd9. )
```

## 10. csacstatus

The csac status command causes CASC to send a test status message to itself from the specified workstation. The message is written by CSAC onto the CSAC message pipe. This command provides a way to test CSAC's response to termination messages from different workstation's and ndd's. The user will be prompted for the originating workstation (workstation or ndd), the overall health of the workstation, the flight 1 identification, flight 1 data stream tag, the flight 1 data quality, the flight 2 identification, flight 2 data stream tag, and the flight 2 data quality.



**Example:**

csacstatus nddl 51-F sim 164

1 1 2 3 4

- 1 - originating workstation

```
(the name of the originating workstation must be in the list of CAS)
(nodes found in the file /cas/application/csac/NET_NAMES. The use )
(of any other name may cause the CSAC process to crash. Presently )
(the node names are: wsl, ws2, nddl, ndd2, ndd3, ndd4 and ndd9. )
```

- ```

2 - overall health
(valid overall health values are: )
(151 - ok )
(152 - marginal )
(153 - in transition )
(154 - failure )
(155 - unknown )

```

- 3 - flight id (

- ```

4 - data stream tag
   (values are:
    (pb1 - playback data dump 1)
    (pb2 - playback data dump 2)
    (rt  - realtime data      )
    (sim - simulation data    )

```

- ```

5 - data quality
  (valid values are: )
  (161 - ok )
  (162 - marginal )
  (163 - unacceptable )
  (164 - normal LOS )

```

\*\*\*\*\* HEALTH and STATUS DEBUG COMMANDS \*\*\*\*\*

1. hprocreport (hproc)      The H&S process report command will
2. hsdebugreport (hsdebug)      The H&S debug report command will
3. hshaltreport (hshalt)      The halt H&S report command will

\*\*\*\*\* SUPERVISOR DEBUG COMMANDS \*\*\*\*\*

1. gmaint (qm)      The queue maintenance command will
2. processcontrol (process)      The display the process control table will display for the user the process control table. No other parameters are needed.

Example:

process

3. supvadvoff
4. sdbug (sdb)
5. osdbug (osdb)

\*\*\*\* USER INTERFACE DEBUG COMMANDS \*\*\*\*

1. announce (an)                      The announce report command will
2. procinit (proci)                  The process initialization command will prompt the user
3. procterm (proct)                  The process termination command will
4. reqscrn (req)                      The request screen command will
5. relscrn (rel)                      The release screen command will
6. ptog (pt)                          The toggle partitions command will
7. report (repo)                      The report view command will bring a report to the screen for viewing. The user will be prompted for the file name. The file name must include the path name. For most reports the path name is:  
/cas/reports

Example:

repo /cas/application/reports/OVERALL

8. uannounce (uan)                  The unannounce report command will

9. udbug (udb)

The debug user interface command will allow the user to enable or disable certain user interface debug options. The user will be prompted for the debug option and if it is to be enabled.

Example:

```
udb 1 1
    ^ |
    | |
    1 2
```

- 1 - debug option  
 (0 - show commands shipped  
 (1 - don't stack advisories, for this option the advisories will)  
 ( appear on the ADVS line, the user will not have to acknowledge)  
 ( them. If there are many advisories they will scroll by. )  
 (2 - show remote TTY command messages  
 (3 - show report queue  
 (4 - show CSAC status  
 (5 - show remote response message

- 2 - enable  
 (0 - No )  
 (1 - Yes)

10. udelay (udel)

The set user interface delay interval command will prompt the user for the number of milliseconds

11. usleep (usl)

The put user interface to sleep command will prompt the user for the number of seconds

## 12. unix

The unix command will allow the user to use unix commands. The user will be prompted for the unix command to be used.

### Example:

```
unix ls -l /cas/log/log_files
```

## 13. rttyenable (rttye)

The enable monitoring of remote tty will allow the user

## 14. rttydisable (rttyd)

The disable monitoring of remote tty command will allow the user

## 15. urelease (urel)

The release screen command will allow the user

## 16. urequest (ureq)

The request screen command will allow the user

## 17. baud

The change remote tty baud rate will allow the user to change the baud rate of the remote tty. The user will be prompted for the baud rate.

### Example:

```
baud 9600
```

18. nice

The nice command will change the priority of a process. The user will be prompted for the application name, the process name, and the adjustment value.

## Example:

```
nice checkout data_generator 10
```

```
  |      |      |
  1      2      3
```

- 1 - application name  
(the values for this parameter are: )  
(support )  
(ndd )  
(ancillary )  
(checkout )
- 2 - process name  
(the values for this parameter are: )  
(supervisor )  
(health\_stat )  
(general )  
(user\_if )  
(ndd )  
(ancillary )  
(data\_generator )  
(score )  
(scriptor )
- 3 - adjustment value  
(The valid values for this parameter are -39 to 39)

APPENDIX A  
CAS COMMANDS

APPENDIX B  
ACRONYMS AND  
ABBREVIATIONS

APPENDIX C  
APPLICABLE  
DOCUMENTATION

## APPENDIX B ACRONYMS AND ABBREVIATIONS

## ACRONYMS AND ABBREVIATIONS

The acronyms and abbreviations used in this manual are as follows:

AB	Abort
A/C	Advisory or Command
ADVS	Advisories
ANCLOG	Ancillary logging
CAS	Calibrated Ancillary System
CCT	Computer Compatible Tape
CNTL	Control
COMMNET	Communications Network
COSW	Checkout Software
CPU	Central Processor Unit
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube (Terminal)
CSAC	Central Status and Control
DARPA	Defense Advance Research Projects Agency
DCP	Data Cyclic Packet
DEV	Developmental, Device
DFE	Data Flow Engineer
DGLOG	Data Generator Logging
DLOG	Delog, Delogging
DOD	Department of Defense
DLOGREQ	Delog Request



## ACRONYMS AND ABBREVIATIONS (Cont.)

EOF	End-of-File
ESC	Escape
F	Function
FACC	Ford Aerospace & Communications Corporation
FLT	Flight
GENLOG	General Logger
GMT	Greenwich Mean Time
GP-LAN	General Purpose Local Area Network
GSFC	Goddard Space Flight Center
H&S, HS	Health And Status
HIST	History
HSLOG	Health and Status Logging
HWCO	Hardware Checkout
ID	Identification
IDD	Interface Definition Document
IEEE	Institute of Electrical and Electronics Engineers
I/F	Interface
ISO	International Standards Organization

# ACRONYMS AND ABBREVIATIONS (Cont.)

JSC	Lyndon B. Johnson Space Center
LAN	Local Area Network
LRU	Line Replaceable Unit
MASSCOMP	Massachusetts Computer Corporation
CORP.	Mission Control Center
MCC	Multiplexer/Demultiplexer
MDM	Master Measurements Data Base
MMDB	Mission Planning
MP	Measurement Stimulus Identification
MSID	Master Timing Unit Increments
MTUI	
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications Network
NCIU	Network Communications Interface Unique
NDD	Network Data Driver
OD	Orbiter Downlink
OI	Operational Instrumentation, Orbiter Instrument
OPS	Operational
OPSDS	Operational Shuttle Data System

## ACRONYMS AND ABBREVIATIONS (Cont.)

P/R	Prompt/Response
QUAL	Quality
REFR	Refresh
RPRT	Report
RT	Real-Time
RT-LAN	Real-Time Local Area Network
SCOLOG	Scoring Logging
SDCP	Simulation Data Cycle Packets
SDSS	STS Data Select Switch
SDT	Shuttle Data Tape
STS	Space Transportation System
SISO	Space Information Systems Operation
S/W	Software

## ACRONYMS AND ABBREVIATIONS (Cont.)

TDT	Telemetry Descriptor Tape
TPC	Telemetry Preprocessing Center
UIF	User Interface
USAF	United States Air Force
UTIL	Utilities
VDT	Video Display Terminal
WS	Workstation

APPENDIX A  
CAS COMMANDS

APPENDIX B  
ACRONYMS AND  
ABBREVIATIONS

APPENDIX C  
APPLICABLE  
DOCUMENTATION

APPENDIX C

APPENDIX C

APPENDIX C

APPENDIX C

## APPENDIX C APPLICABLE DOCUMENTATION

# APPENDIX C APPLICABLE DOCUMENTATION

## Table of Contents

	<u>Page</u>
Standards and Specifications	C-3
Vendor Documentation	C-5

### Standards and Specifications

#### Vendor Documentation

## STANDARDS AND SPECIFICATIONS

### Institute of Electrical and Electronics Engineers (IEEE)

- Local Network Standards IEEE-802

### International Organization for Standardization (ISO)

- Reference Model of Open Systems Interconnection ISO/TC97/SC16/N227

### National Aeronautics & Space Administration (NASA)

- Shuttle Mature Operations Data System (OPSDS) Interface Definition Document (IDD) Standards JSC-10081 Volume II
- CAS System Level B and Subsystem Level B/C Requirements JSC-19227 Volumes I thru IV
- MCC STS and JSC POCC Mature OPS Timeframe Level A Requirements (Rev. B, 8/10/82) JSC-12804
- Safety, Reliability, Maintainability, and Quality Provisions for the Space Shuttle Program NHB-5300.4 (ID-2)
- Inspection System Provisions for Aeronautical and Space System Materials, Parts, Component, and Service NHB-5300.4 (IC)

## STANDARDS AND SPECIFICATIONS (Cont.)

## Ford Aerospace &amp; Communications Corporation (FACC)

- MCC Shuttle Test Plan JSC-10309
- MCC Program General Requirements Specification JSC-10410
- Guidelines for Software Development for Microcomputers Embedded in Hardware Exhibit SISO- EX146
- MCC Cable Separation Criteria for Secure OPS JSC-12229

## Defense Advanced Research Projects Agency (DARPA)

- Department of Defense (DOD) Standard Internet Protocol IEN-128
- DOD Standard Transmission Control Protocol IEN-129
- DOD Standard Trivial File Transfer Protocol IEN-133
- DOD Standard File Transfer Protocol IEN-149
- DOD Space Transportation System Security Classification Guide, USAF Space Division



## VENDOR DOCUMENTATION

- Masscomp UNIX Programming System Model MCS-533
- Televideo Terminal Model 950
- Masscomp 3.0Mb Memory Upgrade Model MEX-530
- Masscomp Floating Point/Array Processor Model FAP-501
- Masscomp 1/4 inch to 1/2 inch Tape Drive Upgrade Model TSX-850
- Masscomp Ethernet Controller Software Model MN-805
- Masscomp Graphics Printer and Cable Model PR-610
- Metacomp Four Channel Input/Output Expander Model MXO-2000-2
- Metacomp RS422 Module Model ELX-2000-200
- Black Box RS232 Interface Switch Model ABC-25 (B-SW020)
- Masscomp 2.0Mb Memory Update Model MEX-520
- Masscomp Menu Development Software SP-35
- Metacomp Intelligent Direct Memory Access Board Model MPA-2000-2